Image-Based Approaches for Photo-Realistic Rendering of Complex Objects

DISSERTATION

zur Erlangung des akademischen Grades

Dr.-Ing.
im Fach Informatik

eingereicht an der Mathematisch-Naturwissenschaftlichen Fakultät II
der Humboldt-Universität zu Berlin

von
Dipl.-Ing. Anna Hilsmann

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:
Prof. Dr. Elmar Kulke

Gutachter(innen):
1. Prof. Dr. Peter Eisert, Humboldt Universität zu Berlin
2. Prof. Dr. Verena Hafner, Humboldt Universität zu Berlin
3. Prof. Dr. Marcus Magnor, Technische Universität Braunschweig

eingereicht am: 11. Oktober 2013
Tag der Verteidigung: 28. März 2014

Abstract

One principal intention of computer graphics is the achievement of photorealism. Although modeling, animation and simulation tools for rendering of complex objects – e.g. human bodies, faces, or clothes – have been developed in the last decades, achieving photorealism by simulating material properties and illumination is still computationally demanding and extremely difficult. This dissertation proposes new approaches for image-based rendering and modification of objects with complex appearance properties, concentrating on the example of clothes. The proposed methods exploit the fact that images are photo-realistic by definition and can be used as appearance examples to guide complex animation or texture modification processes, thereby combining the photorealism of images with the ability to animate or modify an object.

Clothing produces complex shading and texture deformations, which are important for a realistic appearance of the rendered result. For clothing that roughly follows the shape of a human body, it is a reasonable assumption that wrinkling depends on the articulated pose of a human body. Under this assumption, a new image-based rendering approach is proposed, which synthesizes new images of such types of clothing from a database of pre-recorded images based on pose information. Image *warps*, i.e. transformation rules between the images, implicitly extract pose-dependent appearance and shading from the images. For rendering, the images and warps are parameterized and interpolated in *pose-space*, i.e. the space of body poses, using scattered data interpolation.

To allow for appearance changes in image-based methods, a further approach is proposed, which enables image-based *retexturing*, i.e. exchanging the texture or pattern of cloth in an image while maintaining texture deformation and shading properties, without a-priori knowledge of the underlying scene properties. For this purpose, texture deformation and shading are extracted from the input image by estimating an image warp between the input image and an appropriate reference image. In contrast to classical image-based rendering methods, where animation is restricted to viewpoint change and a modification of the object appearance by itself is not possible, the proposed methods allow for complex pose animations and appearance changes.

Both presented approaches build on *warps* between images, not only in the spatial but also in the photometric domain. For this purpose, a new framework for joint spatial and photometric warp optimization is introduced at the beginning of this thesis. This framework estimates mesh-based warp models under a relaxed brightness constancy assumption, which adapts the commonly used brightness constancy assumption to better reflect true conditions of changing and complex lighting conditions between images.

Altogether, the presented approaches shift computational complexity from the rendering to an a-priori training phase. The use of real images and warp-based extraction of deformation and shading allows a photo-realistic visualization and modification of clothes, including fine and characteristic details without computationally demanding simulation of the underlying scene and object properties. This is shown in various rendering, pose synthesis and retexturing results and experiments, which are thoroughly discussed and analyzed.

Zusammenfassung

Fotorealistisches Rendering ist eines der Hauptziele der Computer Grafik. Obwohl in den letzten Jahren zahlreiche Verfahren entwickelt wurden, um komplexe Objekte, wie z.B. Gesichter, menschliche Körper oder Kleidung, realistisch zu modellieren und zu simulieren, ist eine fotorealistische Darstellung durch Simulation von Materialeigenschaften und Umgebungsbeleuchtung immer noch aufwändig und schwierig. Die vorliegende Arbeit entwickelt neue Methoden für Bild-basiertes Rendering von komplexen Objekten sowie deren Modifikation im Bild am Beispiel von Kleidung. Die vorgestellten Methoden nutzen Kamerabilder und deren fotorealistische Eigenschaften für komplexe fotorealistische Animationen und Texturmodifikationen.

Falten- und Schattenwurf von Kleidung sind aufgrund ihrer vielen Freiheitsgrade nur aufwändig zu simulieren. Gerade diese Details sind jedoch essentiell für ein realistisches Rendering Ergebnis. Für eng anliegende Kleidung kann angenommen werden, dass ihr Faltenwurf hauptsächlich von der Pose des Trägers beeinflusst wird. Diese Dissertation schlägt ein neues Bild-basiertes Rendering-Verfahren vor, das unter dieser Annahme neue Bilder von Kleidungsstücken abhängig von der artikulierten Körperpose einer Person aus einer Datenbank von Bildern synthetisiert. Über Abbildungsvorschriften (*Warps*) zwischen den Bildern können Posen-abhängige Eigenschaften, wie Texturverzerrung und Schattierung an Falten, aus den Bildern extrahiert werden. Für die Synthese werden sowohl die Bilder als auch die Warps im sogenannten *Posenraum* parametrisiert und interpoliert.

Um die Erscheinung eines Objekts an sich zu verändern, wird ein weiteres Verfahren vorgestellt, das den Austausch von Texturen in Bildern unter Beibehaltung von Texturverzerrung und Schattierung ohne Kenntnis der zugrunde liegenden Szeneneigenschaften ermöglicht. Hierzu werden Texturdeformation und Schattierung über Bildregistrierung bzw. Warp-Optimierung zu einem geeigneten Referenzbild aus dem Eingangsbild extrahiert. Im Gegensatz zu klassischen Bild-basierten Rendering-Verfahren, in denen die Synthese auf Blickpunktänderung beschränkt und eine Veränderung des Objekts an sich nicht möglich ist, erlauben die vorgestellten Verfahren komplexe Animationen und Texturmodifikation.

Beide vorgeschlagenen Verfahren basieren auf der Kenntnis von örtlichen und photometrischen Abbildungsvorschriften zwischen Bildern. Zu Beginn dieser Dissertation wird ein neues Verfahren vorgestellt, das basierend auf Gitternetz-Modellen örtliche und photometrische Bild-Korrespondenzen schätzt. Dieses Verfahren basiert auf einem angepassten *Brightness Constancy Constraint*, der eine Intensitäts-basierte Fehlerfunktion für komplexe Beleuchtungsänderungen formuliert. Die vorgestellte Methode ermöglicht nicht nur eine simultane Schätzung und kompakte Extraktion von lokaler Deformation und Schattierung, sondern erhöht gleichzeitig die Robustheit gegenüber Beleuchtungsschwankungen zwischen den Bildern. Dies wird in zahlreichen Experimenten demonstriert.

Insgesamt verlagern die vorgestellten Verfahren einen großen Teil des Rechenaufwands von der Darstellungsphase in eine vorangegangene Trainingsphase. Durch den Einsatz von realen Bildern und Warp-basierter Extraktion von Texturverzerrung und Schattierung wird eine realistische Visualisierung und Modifikation von Kleidung inklusive charakteristischer Details ermöglicht, ohne die zugrundeliegenden Szenen- und Objekteigenschaften aufwändig zu simulieren. Dies wird in zahlreichen Experimenten zu Posen-abhängiger Bildsynthese, sowie Texturaustausch gezeigt und analysiert.

Danksagung

Zunächst möchte ich Prof. Dr. Peter Eisert für die Möglichkeit danken, Teil seiner Gruppe am Fraunhofer HHI und an der HU Berlin zu werden. Ganz besonders danke ich Peter für seine fachliche Anleitung, sein Vertrauen und die Freiheiten, die er mir in den letzten Jahren entgegengebracht hat, und nicht zuletzt für die ausgesprochen angenehme Arbeitsatmosphäre, die er geschaffen hat.

Prof. Dr. Marcus Magnor und Prof. Dr. Verena Hafner danke ich für ihr Interesse an meiner Arbeit und ihre Bereitschaft, als Gutachter für die vorliegende Dissertation tätig zu sein.

Allen ehemaligen und aktuellen Mitgliedern der Computer Vision & Graphics Gruppe am Fraunhofer HHI danke ich für die harmonische und fröhliche Arbeitsatmosphäre und die vielen anregenden fachlichen Diskussionen, die oft zu neuen Ideen führten: Peter, Jo und Markus, mit denen ich in den letzten Jahren ein Büro teilen durfte – Danke für inspirierenden Gespräche und auch die vielen kleinen Lacher zwischendurch –, Benjamin, Philipp, Wolfgang, Daniel, David und Christoph. Ich kann mir keine besserern Kollegen vorstellen und das Arbeiten in dieser Gruppe hat mir immer Spaß gemacht. Benjamin und Jo bin ich besonders dankbar für die aufopfernde Hilfe bei der Aufnahme der Daten und Erstellung von Videomaterial vor der Eurographics Deadline.

Danke an meinen Vater Jo und Henning für das sorgfältige Korrekturlesen der Arbeit.

Meiner Familie und meinen Freunden möchte ich für die anhaltende Motivation, Ermutigung sowie für die entgegengebrachte Nachsicht danken.

Mein größter Dank gilt Henning - für alles, ganz besonders aber für seine unermüdliche Geduld, Toleranz und anhaltende Ermutigung.

Contents

List of Figures

List of Tables

Nomenclature

Basic Notations

$\mathbf{0}_{n\times m}$	$n \times m$ matrix with all elements 0
$\mathbf{1}_{n\times m}$	$n \times m$ matrix with all elements 1
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$
$f(\mathbf{x})$	Function in $\mathbf{x}$
$f(\mathbf{x};\boldsymbol{\theta})$	Function in $\mathbf{x}$, parameterized by a parameter vector $\boldsymbol{\theta}$
$\mathbf{g}_f$	$n \times 1$ gradient vector of $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{g}_f = \nabla f$
$\mathbf{H}_f$	$n \times n$ Hessian matrix of $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{H}_f = \nabla^2 f$
$\mathbf{I}_n$	$n \times n$ Identity matrix
$\mathbf{J_f}$	$m \times n$ Jacobian matrix of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$
x	Scalar
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\mathbf{x}^T, \mathbf{X}^T$	Transpose of a vector or matrix
∇	Nabla operator, $\nabla = [\partial/\partial x_1 \ \ldots \ \partial/\partial x_n]^T$
Δ	Laplace operator $\Delta = \sum_{k=1}^{n} \partial^2/\partial x_k^2$
$\circ$	Hadamard or entrywise matrix product $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$ with $c_{ij} = a_{ij} \cdot b_{ij}$
$\|.\|$	L^2-norm, Euclidean norm, $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$

Vectors and Matrices

Elements of a vector or matrix are usually denoted with subscripts:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a_{11} & \ldots & a_{1N} \\ & \ddots & \\ a_{N1} & \ldots & a_{NN} \end{bmatrix}$$

$\text{diag}\left[\mathbf{A}_1 \dots \mathbf{A}_n\right]$	Block diagonal matrix built from the matrices $\mathbf{A}_1 \dots \mathbf{A}_n$: $$\text{diag}\left[\mathbf{A}_1 \dots \mathbf{A}_n\right] = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{bmatrix}$$
$\text{diag}_n[\mathbf{A}]$	Block diagonal matrix built from n-times the matrix $\mathbf{A}$: $$\text{diag}_n[\mathbf{A}] = \begin{bmatrix} \mathbf{A} & & \\ & \ddots & \\ & & \mathbf{A} \end{bmatrix}$$
$\text{diag}(\mathbf{A})$	Diagonal matrix built from the diagonal elements of $\mathbf{A}$: $$\text{diag}(\mathbf{A}) = \begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{NN} \end{bmatrix}$$
$\text{diag}(\mathbf{x})$	Diagonal matrix built from the elements of $\mathbf{x}$: $$\text{diag}(\mathbf{x}) = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_N \end{bmatrix}$$

Common Quantities

The notation in this thesis is consistent such that common quantities are always referred to by the same symbols. The following is a list of all symbols commonly (but not exclusively) used to denote the same throughout the thesis. Symbols not in this list are directly explained in the context.

$\mathbf{B}_s(\mathbf{x}_i)$	Parametrization matrix for the spatial warp
$\mathbf{B}_p(\mathbf{x}_i)$	Parametrization matrix for the photometric warp
$\mathcal{E}(\boldsymbol{\theta})$	Cost function in the warp optimization framework
$\mathcal{E}_D(\boldsymbol{\theta})$	Data term of the cost function
$\mathcal{E}_S(\boldsymbol{\theta})$	Smoothness term of the cost function
$\mathcal{I}(\mathbf{x})$	Image intensity or color at pixel position $\mathbf{x}$. A grayscale image can be considered as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}$, mapping a pixel coordinate onto an intensity value. A color image can be considered as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}^3$, mapping a pixel coordinate onto a RGB color vector.
$\mathcal{I}^R, \mathcal{I}^G, \mathcal{I}^B$	Red, green and blue channels of an image
$\nabla\mathcal{I}$	Image gradient, $\nabla\mathcal{I} = [\mathcal{I}_x\ \mathcal{I}_y]^T = \left[\frac{\partial\mathcal{I}}{\partial x}\ \frac{\partial\mathcal{I}}{\partial y}\right]^T$
$\mathbf{K}$	Intrinsic camera matrix
$\mathbf{L}$	Laplace matrix of a mesh

$\mathcal{M}$	Mesh $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ consisting of a set of vertices $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_K]^T$ and a set of faces $\mathcal{F}$
$\mathcal{N}_k$	Neighborhood of a vertex with index $_k$ in a mesh or graph
$\mathcal{R}$	Image region
$\mathbf{q}$	Pose parameterization vector, e.g. a vector of joint angles or joint locations
$\mathbf{V}$	Matrix of concatenated mesh vertices $\mathbf{V} = [\mathbf{v}_1 \ \dots \ \mathbf{v}_K]^T$
$\mathbf{v}$	Mesh vertex $\mathbf{v} = [u \ v]^T$ or $\mathbf{v} = [u \ v \ w]^T$. The dimension is given in the context. Vertices $\mathbf{v}_k$ are indexed in the range $k = 1 \dots K$ and can be identified by their index.
$\mathcal{W}$	Joint spatial and photometric image warp
$\mathcal{W}_{i \to j}$	Warp between two images $\mathcal{I}_i$ and $\mathcal{I}_j$
$\mathcal{W}_p$	Photometric warp $\mathcal{W}_p : \mathbb{R}^{N_p} \to \mathbb{R}$
$\mathcal{W}_{pc}$	Photometric warp for color images $\mathcal{W}_{pc} : \mathbb{R}^{N_{pc}} \to \mathbb{R}^3$
$\mathcal{W}_s$	Spatial warp $\mathcal{W}_s : \mathbb{R}^{N_s} \to \mathbb{R}^2$
$\mathcal{W}^{\mathcal{D}}$	Detail-warp in the Pose-Space Image-Based Rendering framework
$\mathcal{W}^{\mathcal{SSD}}$	SSD-warp in the Pose-Space Image-Based Rendering framework induced by SSD animation with LBS skinning
$\mathbf{x}$	Pixel coordinate $\mathbf{x} = [x \ y]^T$. Pixels $\mathbf{x}_i$ are indexed in the range $i = 1 \dots P$ and can be identified by their indices.
β	Barycentric coordinate
$\mathbf{\Gamma}$	Tikhonov regularization matrix
$\gamma(\boldsymbol{\theta}_\gamma)$	Color gain function of the photometric warp
γ_{bg}	Global blue gain parameter
γ_{rg}	Global red gain parameter
λ	Regularization parameter
$\delta\boldsymbol{\theta}$	Parameter update in the optimization framework
$\boldsymbol{\theta}$	Joint spatial and photometric warp parameter vector
$\hat{\boldsymbol{\theta}}$	Estimated warp parameter vector
$\boldsymbol{\theta}_p$	Photometric warp parameter vector
$\boldsymbol{\theta}_s$	Spatial warp parameter vector
$\boldsymbol{\theta}_\gamma$	Red and blue intensity gain parameters $\boldsymbol{\theta}_\gamma = [\gamma_{bg} \ \gamma_{rg}]^T$
ρ_k	Photometric parameter of vertex $\mathbf{v}_k$
ψ	General kernel function in the optimization framework
ψ_{H}	Huber kernel
ψ_{LS}	Least-squares kernel

Acronyms

2D	2-dimensional
3D	3-dimensional
fps	frames per second
GN	Gauss-Newton
IBR	Image-Based Rendering
ICP	Iterative Closest Point
kNN	k-Nearest Neighbors
L^2	L^2-norm, Euclidean norm
LBS	Linear Blend Skinning
LM	Levenberg-Marquardt
LS	Least-Squares
MAD	Median Absolute Deviation
NRT	Near-Regular Textures
PSD	Pose-Space Deformation
PS-IBR	Pose-Space Image-Based Rendering
RBF	Radial Basis Function
RGB	Red Green Blue
RMSE	Root Mean Squared Error
SIFT	Scale Invariant Feature Transform
SSD	Skeleton Subspace Deformation
s	Optimization of a spatial warp
sp	Optimization of a joint spatial and photometric warp
spc	Optimization of a joint spatial and photometric warp for color images

1. Introduction

1.1. Motivation and Objectives

Ever since the beginning of computer graphics, modeling and simulation tools have been developed to create realistic and compelling visual content of complex objects, such as human bodies, faces or clothes. Achieving real photorealism for such real-world objects with these tools, however, is still difficult and computationally demanding, as it requires sophisticated simulation of material properties and illumination. This dissertation addresses a photo-realistic visualization of objects with complex appearance properties, concentrating on the example of clothes. Visualizing clothes is still a challenging problem, because cloth deformation and drapery exhibit many degrees of freedom and wrinkles produce complex shading and texture deformations. Yet, these complex details are essential for a realistic appearance of the rendered clothes. Usually, the visualization of clothes relies on a textured 3-dimensional (3D) computer graphics model, and foldings as well as dynamics of the cloth are synthesized. Traditionally, the parameters for the synthesis process are calculated with physics-based methods [CK05, GHF+07, VMTF09]. Recently, many methods have been proposed that are directly inspired by the *real world* and learn deformation and reflection parameters of clothes for physical simulation from visual and other sensors [BTH+03, BPS+08, PZB+09, MBT+12]. Although compelling results can be achieved with these methods, the simulation is still computationally demanding because of the above mentioned complex properties of cloth. It is therefore difficult and time-consuming to simulate the underlying physical properties, e.g. 3D geometry, deformation, self-collision, light sources, physical shading and reflection properties, accurately.

An alternative to synthesis and reconstruction is observation of *appearance* through a number of images. Images are photo-realistic by definition. However, as images are static, animation and modification of an object captured in the images, is not possible at first. This is addressed by image-based rendering (IBR) approaches, which aim at combining the photorealism of images with the ability to modify or animate the scene content to some degree. These methods synthesize new images by appropriately interpolating and merging a database of prerecorded images [LH96, BBM+01, DLD12]. Typically, the database images show various viewpoints of a rigid and stationary object or scene – and can therefore be seen as view-dependent *appearance examples* – and the synthesis of new images is limited to viewpoint changes. Complex scene lighting, shading and reflection properties during viewpoint change can realistically be rendered with these methods without

simulating the underlying scene properties. However, changing the appearance of an object, e.g. to exchange the texture, or more complex animations than viewpoint change are generally not possible. The appearance of a piece of clothing, for example, not only depends on viewpoint but also on the pose of a person wearing it. The main idea of this dissertation is to approach the task of clothing visualization and modification (animation and texture replacement) in an image-based manner, using as much information from real images as possible. This approach exploits the fact that all characteristics, such as texture deformation and shading properties at fine wrinkles, are implicitly captured by the images. This information can be extracted from the images as *warps* between them, i.e. transformation rules that map one image onto another. These warps are not only extracted in the spatial domain, i.e. as a deformation field, which moves the pixels from one location to another, but also in the photometric domain, changing the intensities of pixels, thereby allowing simultaneous capturing of local deformation and shading. Depending on the nature of the images, the extracted warps can then be exploited in several ways:

- Under the assumption that the appearance of tight-fitting clothing, e.g. trousers or shirts, is pose-dependent [WHRO10], a database of images showing a piece of clothing in different body poses and from different viewpoints can be regarded as a database of pose-dependent appearance examples of this piece of clothing. Warps between the database images yield information on pose-dependent texture and shading changes between poses. The database images and warps can then be mapped onto an appropriate space, e.g. that of all body poses, to synthesize new clothing images as a function of body pose in a pose-dependent image-based rendering approach.
- A warp to an appropriate reference image of an undeformed and uniformly lit texture yields information about absolute texture deformation and shading properties, which can be exploited for *retexturing*, i.e. the replacement of the original texture by a synthetic one while maintaining texture deformation, shading and lighting conditions from in the original image.

Summing up, the main idea of this work is to approach the task of clothing visualization in an image-based manner, exploiting the fact that images provide information on very characteristic properties of the appearance of clothing, such as wrinkling and shading, which are very important for a photo-realistic visualization and difficult to simulate. To capture these characteristics from the images, the idea is that texture deformation and shading can be extracted as spatial and photometric image warps. The use of real images and warp-based extraction of deformation and shading will allow a photo-realistic and plausible example-based visualization and retexturing of clothes. The main incentive for this approach is that the aim is rather a plausible, photo-realistic and perceptually correct visualization than physically accurate and correct reconstruction.

Applications. The applications of the methods presented in this dissertation are manifold. The main targeted application is the visualization of clothes in augmented reality applications such as virtual try-on of clothes (Fig. 1.1), called *Virtual Mirror*

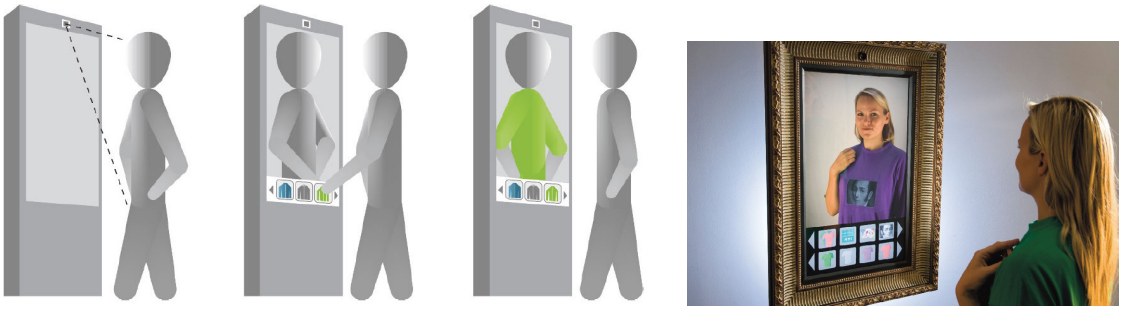


Figure 1.1.: Left: concept of a Virtual Mirror application. A user can select between various clothings on a touchscreen and can then see his reflection in the mirror augmented with the virtual clothing while moving freely in front of the system. Right: realized Virtual Mirror prototype.

in the following. In such an environment, a user is tracked by one or more cameras and visualized on a display, showing his/her reflection in the mirror augmented with new clothes, which can be changed and configured by the user. Such, a user or designer can try arbitrary styles of clothes before production and can configure it for his/her special needs. There has been a huge increase in virtual try-on applications in recent years. The first available systems show a static image of the piece of clothing chosen by the user, and the user has to position himself such that the rendered images of the clothes roughly fit his shape[1]; or they display physically simulated clothes on virtual avatars [DTE+04, MTKV+11], thus limiting the usefulness and realistic experience for the user. With recent development in video plus depth sensors and human body tracking methods (e.g. Microsoft Kinect [SFC+11]), virtual try-on systems have been released lately, which render virtual clothes onto the user's body in real-time[2]. The photo-realistic visualization of the clothes in real-time, however, remains the bottleneck in these systems, and the clothes still appear unrealistic and computer-generated.
Further possible applications would be in movies or video games where sophisticated image-based rendering and retexturing methods for clothing would allow media creators for films and games to generate, modify and manipulate clothes effectively or to create photo-realistic avatars.

1.2. Contributions

The principal contributions of this dissertation are:

- A new framework for **Joint Spatial and Photometric Image Warp Optimization.** Typically, image registration refers to estimating a spatial transfor-

[1] Try On Bathing Suit: http://www.tryonbathingsuit.com
Imagine That: http://www.imaginethattechnologies.com
Web Cam Social Shopper http://webcamsocialshopper.com
[2] Fitnect: http://www.fitnect.hu

mation between two images. Intensity-based methods optimize an error function formulated over the image intensities. Most state-of-the-art approaches treat intensity differences between images as outliers and try to compensate for these differences e.g. through robust estimation. In contrast to that, the proposed approach explicitly models intensity differences between images in a photometric warp in addition to a spatial warp (Chapter 3). These warps register two images not only spatially but also photometrically, i.e. in the intensity domain. A framework for the joint estimation of these warps is proposed based on a relaxed brightness constancy assumption and mesh-based warp models, allowing for non-rigid deformation and local brightness changes. Joint estimation of the warps not only makes the optimization more robust against intensity changes but at the same time allows actual *extraction* of intensity differences between images in addition to local deformation. The extracted information can for example be exploited to synthesize new images in image-based rendering or retexturing approaches. In general, the presented framework allows the incorporation of generic warp models because it is formulated in a modular manner such that the warp models can be formulated according to the given problem. The knowledge of spatial and photometric warps between images is essential for the image-based approaches presented in this dissertation.

- A new approach for **Image-Based Rendering of Articulated Objects with Complex Pose-Dependent Appearance.** In contrast to traditional image-based rendering approaches, which are limited to viewpoint change or temporal interpolation, the proposed approach accounts for pose-dependent appearance to synthesize new images as a function of body pose (Chapter 4). A coarse geometric model allows low-resolution shape adaptation, e.g. animation and view interpolation, whereas small details as well as complex shading and reflection properties are modeled by pose-dependent appearance examples (images) stored in a database. Correspondences between the images are represented by mesh-based image warps, both in the spatial as well as in the intensity domain. A parameterization of pose positions the examples at scattered positions in *pose-space*, i.e. the space of possible body poses. For rendering, the stored warps as well as intensities are interpolated by exploiting scattered data interpolation methods that have already been successfully used in example-based animation. The high dimensionality of the interpolation domain is tackled by partitioning the pose-space into subspaces related to different parts of the body.

- A new framework for **Image-based Retexturing**. The proposed approach allows to exchange the texture of a surface or object in an image without knowledge or explicit reconstruction of the scene properties, such as 3D shape or lighting conditions. The key motivation is that texture distortion and shading represent strong cues for the perception of shape such that 3D geometry information is not needed (Chapter 5). Given a reference of the undeformed and uniformly lit texture, these properties can directly be extracted from the

image as a joint spatial and photometric warp. A virtual texture can then be warped and rendered into the image such that the original deformation and shading are maintained. Because such a reference image is often difficult to provide, a method for a specific type of textures (near-regular textures (NRT)) is presented, which estimates the appearance of the reference texture in a texture analysis step prior to warp optimization.

1.3. Overview

This dissertation is structured as follows.

Chapter 2 Related Work: Chapter 2 reviews the state of the art in Computer Vision and Computer Graphics fields related to the methods presented in this thesis, i.e. non-rigid image warp estimation (Sec. 2.1), image-based rendering (Sec. 2.2), pose synthesis and interpolation (Sec. 2.3), and texture replacement in images (Sec. 2.4). Finally, an overview on available virtual try-on systems is given (Sec. 2.5).

Chapter 3 Joint Spatial and Photometric Warp Optimization: Chapter 3 introduces a framework for joint spatial and photometric image warp estimation with mesh-based models [HE08, HE09a, HSE10]. It starts with a background on spatial image warp optimization based on the brightness constancy assumption and mesh-based models (Sec. 3.1). Sec. 3.2 proposes an extension to joint spatial and photometric warp estimation (Sec. 3.2.1) and an extended warp formulation for color images (Sec. 3.2.2). The optimization framework is summarized in Sec. 3.2.3. Finally, Sec. 3.3 evaluates and analyses the proposed approach. Various applications are presented throughout this thesis.

Chapter 4 Pose-Space Image-Based Rendering: Chapter 4 introduces an image-based rendering approach for articulated objects with complex pose-dependent appearance, such as clothes [HE12, HFE13]. The proposed image-based representation and the concept of parameterization in pose-space is detailed in Sec. 4.1: a definition of the representation in Sec. 4.1.1 is followed by a detailed description of the database generation in Sec. 4.1.2. Sec. 4.2 focuses on the synthesis of new pose images from this database: scattered interpolation of example warps is addressed in Sec. 4.2.1; blending and merging of warped database images is described in Sec. 4.2.2; Sec. 4.2.3 focuses on the partition of the interpolation domain into subspaces related to different parts of the body; a distance measure to select database images for a smooth animation is presented in Sec. 4.2.4. Finally, Sec. 4.3 discusses experiments and results.

Chapter 5 Image-Based Retexturing: Chapter 5 proposes an image-based retexturing approach, exploiting joint spatial and photometric warps between an image and an appropriate reference texture [HSE11a, HSE11c]. After a short introduction on the idea of warp-based retexturing, Sec. 5.1 presents the theory of *near-regular textures*, which is used to estimate the appearance of an appropriate, i.e. unwarped,

reference image from a single image, as proposed in Sec. 5.2. Sec. 5.3 presents various retexturing results.

Chapter 6 Conclusions: Finally, Chapter 6 concludes this dissertation and provides an outlook on how to apply, extend and combine the approaches presented in this thesis.

1.4. Research Publications

In accordance with Section 6, Article 2b) of the doctorate regulations of the faculty of natural sciences at the Humboldt University of Berlin, parts of this thesis have been presented at the following international conferences and workshops

- Eurographics 2011 [HSE11a], 2012 [HE12] and 2013 [HFE13];
- Vision, Modeling and Visualization Workshop 2009 [HE09b] and 2011 [HSE11c];
- Mirage 2009–International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications [HE09c];
- CVPR/ICCV Workshops on Non-Rigid Shape Analysis and Deformable Image Alignment 2008 [HE08] and 2009 [HE09a];

or has been published in Computers & Graphics vol. 34(5) [HSE10]. These publications are the foundation of this thesis, which incorporates them under the presented approaches for image-based clothing visualization and presents an enhanced analysis of the approaches, together with updated results and discussions.

2. Related Work

This chapter reviews existing work and state-of-the-art approaches in research fields related to this thesis.

2.1. Non-Rigid Image Warp Estimation

A huge amount of literature exists on deformable image registration and warp estimation, and this section can only give a compact overview on approaches related to this field. For more detailed surveys, the reader is referred to e.g. [BFB94, ZF03, BM04, SDP13].

Correspondences between two or more images, e.g. images in a multi-view camera setup or images in a time-sequence, are key to many computer vision tasks, such as 3D reconstruction [SHE11b, BPS+08], deformable motion tracking [PLF05a] or medical image analysis [SHE11c]. Typically, two images are considered, one of which is referred to as source image $\mathcal{I}_{\mathcal{S}}$, i.e. the image that is *warped*, and the other one is referred to as target image $\mathcal{I}_{\mathcal{T}}$. The correspondences between the two images are commonly described by a dense displacement vector field, which links the location of each pixel in the source image to its location in the target image. During registration, the source image undergoes a transformation or *warp* and the goal of registration is to find the warp that best aligns the source image with the target image. Basically, warp estimation involves three components: (i) the definition of the warp model, i.e. the parameters to be optimized, (ii) the definition of the objective function to be minimized during optimization and (iii) the optimization method.

The type of the warp model defines the number of parameters optimized during warp estimation and often implies an assumption on the nature of the deformation field. The number of degrees of freedom can range from e.g. 6 for an affine transformation up to twice the number of pixels for a dense displacement field. The higher the degrees of freedom, the more complex deformations can be described by the model, but the more challenging is the estimation. Deformable warp models often interpolate or approximate a dense displacement field from sparse positions or control points in the image. Popular non-rigid warp models include radial basis functions (RBF) [Boo89, BZ04, LY05, YXLX11], freeform deformation [KU03] or deformable triangle meshes, i.e. piecewise affine warps [GBBS10, ZLMH09]. The type of the RBF as well as the number of control points or mesh vertices determine overall characteristics of the transformation such as the smoothness or the locality.

Regarding the objective function, the literature basically distinguishes between marker-based [SM06, WCF07], feature-based [Low03, BETG08, MY09, PLF08] and intensity-based [Hor86, BA93, IA99, BZ04, BBPW04, LY05, XJM10] cost functions. As markers are not always available in real scenes, the assumption of such a-priori knowledge limits the applicability of these methods to special cases. Feature-based methods minimize a distance between sparse sets of corresponding feature points, and a dense displacement field is interpolated from these distinct positions. Finally, intensity-based methods minimize an error measure based on intensity information of all pixels in the image.

Feature-based methods determine correspondences between distinct and salient feature points in the images and use them to estimate a transformation that best matches the two feature point sets. As this thesis mainly concentrates on intensity-based warp estimation methods, this section only gives a brief overview of the most important feature detectors that have been described in the literature. Image-based feature point detectors include local curvature extrema or saddle points, edges or corners [Thi96, HS88] or scale-space features like *SIFT* [Low03], *ASIFT* [MY09] and *SURF* [BETG08] features. Typically, two sets of feature points $\mathbf{P}_{\mathcal{S}} = \{\mathbf{p}_{\mathcal{S}_1} \dots \mathbf{p}_{\mathcal{S}_N}\}$, $\mathbf{P}_{\mathcal{T}} = \{\mathbf{p}_{\mathcal{T}_1} \dots \mathbf{p}_{\mathcal{T}_M}\}$ are created in the source and the target images. One approach to estimate a warp from these feature points, is to first establish correspondences between the two point sets and then estimate a transformation while eliminating outliers. To determine correspondences, a common approach is to search for nearest neighbors in descriptor space under some consistency constraints, i.e. non-ambiguity and consistency in both matching directions [Low03, BETG08, MY09]. A popular method to eliminate outliers during the estimation of the transformation is the Random Sample Consensus *RANSAC* algorithm [FB81]. It is widely used to compute transformations with a low degree of freedom, but for more complex non-rigid ones its complexity grows exponentially with the degrees of freedom of the transformation model. Consequently, feature-based methods are mostly used to find global relations between images. If the set of points is large enough, more complex transformations, e.g. deformable meshes, can be determined by minimizing descriptor differences with a robust estimator to eliminate outliers [PLF08].

In contrast to feature-based approaches, intensity-based methods do not rely on sparse features but instead fit the warp model directly to the image data, mainly by exploiting the brightness constancy assumption between the source and the target images $\mathcal{I}_{\mathcal{S}}$ and $\mathcal{I}_{\mathcal{T}}$ [Hor86, IA99]. This equation is given by

$$\mathcal{I}_{\mathcal{S}}(\mathbf{x} + \Delta\mathbf{x}) - \mathcal{I}_{\mathcal{T}}(\mathbf{x}) = 0 \ ,$$

where $\mathbf{x} = [x\ y]^T$ denotes a pixel location in the image and $\Delta\mathbf{x} = [\Delta x\ \Delta y]^T$ denotes its displacement. The brightness constancy equation assumes that differences between the two images can be fully explained by a pixel displacement field, which moves a pixel from one location to another, whereas the intensities in the image do not change. The first Taylor expansion of the brightness constancy assumption is often called the *Optical Flow* equation:

$$\mathcal{I}_{\mathcal{S}}(\mathbf{x}) - \mathcal{I}_{\mathcal{T}}(\mathbf{x}) + \nabla\mathcal{I}_{\mathcal{S}}^T \cdot \Delta\mathbf{x} = 0 \ .$$

Most intensity-based warp estimation methods optimize an error function based on the brightness constancy assumption using non-linear optimization techniques. This error function is often the sum of squared pixel errors (least-squares optimization) [Hor86] or a more robust error functional [BA93, BBPW04], but other measures like normalized cross correlation have also been used [BFB94].

Generally, image-based warp estimation is an ill-posed problem, and a regularization is needed because the number of parameters for a dense pixel displacement field is larger than the provided constraints. A predefined warp model with fewer parameters, e.g. a mesh-based model, implicitly provides additional constraints for regularization (this type of regularization is therefore often called *internal* or *implicit regularization*). Another way of regularization is to add smoothness constraints to the objective function (often called *external* or *explicit regularization*). Besides making the optimization problem well-posed, both regularization types can be used to incorporate prior knowledge on the smoothness of the deformation. Common smoothness constraints typically penalize the spatial second derivative of the displacement field [HS81]. Recently, more and more methods use a regularization term based on the total variation of the displacement field, which allows for sharp discontinuities in the warp [BBPW04, PSG$^+$08, WPZ$^+$09, ZGK$^+$10, BM11]. However, total variation regularization often suffers from the so-called *staircasing* effect, i.e. piecewise constant solutions [PCBC10]. For this reason, in practice, often a quadratic penalization is used for small values of the error term and a linear penalization for larger values, which is essentially the Huber norm [PCBC10], having its origins in robust statistics [Hub81].

There are two main difficulties that make the optimization in an image-based approach challenging: (i) intensity information can lead to ambiguous matching such that the cost function might not be convex and exhibit local minima, (ii) robustness against outliers that validate the brightness constancy assumption (e.g. appearance variations due to lighting changes). To overcome the first, many methods require an initial guess close to the global minimum. This has been approached e.g. by an initialization of the warp optimization with parameters estimated from corresponding feature points [XJM10] or a combination of intensity-based error terms with descriptor-based terms [ZLMH09, BM11]. Other approaches use a multi-scale approach [LK81, BBPW04] and/or a hierarchy of deformation models [BAHH92]. Recently, methods have been proposed that develop a convex formulation of the originally non-convex objective function by lifting the optimization problem to a higher dimensional space [PSG$^+$08, PCBC10], thereby guaranteeing finding the global optimum.
Outliers in the data have been accounted for by using robust estimators [BA93, BBPW04] in the cost function of the optimization framework. These estimators limit the influence of outlying data in the cost function. Appearance variations caused by lighting changes in particular have been accounted for in different ways. One approach is to use photometric invariants, e.g. photometric invariant color spaces [GB97, vdWG04], spherical/canonical transformations [vdWG04, MBW07] and/or normalization strategies [MBW07]. Also, cost functions based on gradient values

instead of intensity values [BBPW04] have been used. However, gradient-based cost functions are only invariant against additive illumination changes. Another approach is to perform a global lighting adaptation after each optimization step [SHE11a]. Also, error measures like cross correlation are more invariant against multiplicative intensity changes than the sum of squared pixel errors [MCF10]. Preprocessing, like separating the images into reflectance and shading [STL08] or structure and texture components [WPZ+09] prior to warp estimation, can make the estimation more robust against lighting changes. However, separating images in such components is another ill-posed problem.
While these approaches try to compensate for intensity differences between images, there have also been some approaches to explicitly allow appearance changes in the warp model [GN87, TLCH02, HF01, Bar08, SM07]. Gennert and Negahdaripour [GN87] were among the first to propose an intensity-based method robust to illumination variations by explicitly allowing brightness changes. They assumed that the brightness in an image taken at time $t + \delta t$ is related to the brightness in a second image taken at time t through a set of parameters, which can be estimated from the images. A similar method was later proposed by Teng et al. [TLCH02]. Several other researchers [HF01, Bar08, SM07] have exploited their ideas to make image registration more robust against lighting changes. Some approaches jointly estimate optical flow and the parameters of a physical light source causing illumination changes [HF01, EG97, EG02]. In these cases, the parameters are restricted to the assumed global lighting conditions. Other methods directly estimate a global intensity or color transformation between the images to compensate intensity differences between the images [SM07, Bar08]. In these methods, the additional estimation of photometric parameters is exploited to make spatial image registration more robust against intensity differences between the images. However, actual *retrieval* of local shading patterns together with local non-rigid deformation is not addressed.

The optimization method to minimize the objective function generally depends on the problem formulation. Continuous methods are constrained to a differentiable objective function and real-valued variables. This is often the case in warp optimization problems, and iterative methods, such as Gradient Descent, Gauss-Newton or Levenberg-Marquardt have mainly been used [GBBS10, ZLMH09, PCBC10]. These methods perform a local search and are thus sensitive to initial conditions if the objective function is not convex. Discrete methods, on the other hand, perform a global search but are limited to a quantized search space and thus tend to be more inaccurate. In this category, amongst others, graph-based methods have been used [TC07, GKT+08, ZGK+10].

2.2. Image-Based Rendering and Interpolation

Generating new images by warping and merging existing images from a database, as proposed in Chapter 4, is closely related to *image-based rendering* and *image interpolation*. This section focuses on *classical*, i.e. view-dependent, image-based ren-

dering techniques (Sec. 2.2.1) and spatio-temporal image interpolation (Sec. 2.2.2). Image-based pose synthesis and animation is covered by Sec. 2.3.

2.2.1. Image-Based Rendering

Image-based rendering (IBR) techniques render new virtual views of an object directly from a pre-recorded set of images, in some cases exploiting additional geometry or depth information. This is different from traditional 3D computer graphics rendering, where a single geometric model with a single texture is used and view-dependent appearance has to be modeled explicitly by physically modeling light sources, scattering of light rays etc. In image-based rendering, view-dependent appearance is presented by a database of *examples*, i.e. images of an object taken from different viewpoints, and new viewpoint images are interpolated from this database [LH96, MB95, BBM+01, DLD12]. Thereby, view-dependent properties, e.g. complex reflection and shading, can be rendered without simulating the underlying scene and objects properties even for objects with complex refraction and reflection properties [EJH10]. Shum and Kang [SK00] classify image-based rendering techniques based on how much information about the scene geometry is used to warp and blend the pre-captured images. Basically, the amount of images and geometry is a trade-off between photo-realism and compression of the model. Pure image-based representations have the advantage of photo-realistic rendering but require a dense sampling of the input view space with a large number of images. Hence, they have high costs of data acquisition and storage requirements. An estimation of the scene geometry (a geometric *scene proxy*) is often used to reduce the number of images and to synthesize new views from sparser datasets [DTM96, PDG05, ER06] with the limit of a single 3D model with a static texture [CSN07].

The earliest works in image-based rendering, like *Light Fields* [LH96] or *Plenoptic Modeling* [MB95], did not use any 3D geometry information about the scene. These methods generate new images by appropriately filtering and interpolating rays from a pre-captured set of images. Methods such as *Lumigraph Rendering* [GGSC96], *View-Dependent Texture Mapping* [DTM96, PDG05] or *Voxel Coloring* [SD97] use approximate depth maps or geometric models to create new views from fewer images. Unstructured input camera positions and viewpoints have been addressed both for a generalized Lumigraph approach [BBM+01] as well as for an unstructured version of the Lightfield approach [DLD12]. Depth information is often either provided by z-cameras or multi-view stereo (since an overview on multi-view stereo methods is out of the scope of this thesis, please see [SCD+06, Mida, Midb] for an overview). However, an accurate generation of scene geometry remains difficult in practice such that approaches relying on accurate scene geometry can suffer from blurring and ghosting artifacts. For this reason, methods have been proposed to reduce those artifacts on texture-basis, e.g. by warp-based alignment of the textures [EDM+08, DMC+12] or by globally optimizing a texture assignment problem [LI07, GWO+10].

Image-based rendering has been extended to free-viewpoint video, often specialized to the rendering of humans [CTMS03, ZKU+04, LLB+10]. Also, representations based on billboards have been specifically designed for the visualization of human actors [WWG07, BBPP10, GHK+10]. These methods are restricted to viewpoint changes; changes in motion, however, e.g. geometric modification or animation of the human actors, are not possible. Pose-dependent image-based rendering and animation of humans is covered by Sec. 2.3.

2.2.2. Image Interpolation

Image interpolation transforms two images of the same scene (separated in space or time) to generate in-between images with a seamless transition [SD96, LLB+10, SLW+11]. This is generally done by determining dense or sparse image correspondences and gradually warping both images onto each other to compensate for differences in object pose or viewpoint before blending. This requires a small baseline or small motion between the two images. An optical flow-based space-time interpolation method has been presented in [VBK02], restricted to interpolation between views and consecutive video frames. Stich et al. [SLW+11] presented a perceptually motivated image interpolation method implementing concepts of human vision. The approach is based on the observation that e.g. small discontinuities at image edges during a motion are more disturbing to the human vision system than a smooth yet physically incorrect motion. Motivated by this fact, the approach aims at computing images that are visually convincing rather than interpolating the measured optical flow. Recently, a method for complex real-world scenes from uncalibrated multi-video footage was proposed by Lipski et al. [LLB+10]. In their method, the captured images are parameterized in a 3D space-time interpolation domain and a tetrahedralization of that space is used for interpolation with known correspondences between images connected by an edge. Their space-time parameterization allows for spatial viewpoint navigation, slow motion, or freeze-and-rotate effects.

2.3. Pose Synthesis

This thesis addresses the problem of synthesizing an image of a piece of clothing based on a given articulated pose of a human body (Chapter 4). This is approached in an image-based manner. Image-based pose synthesis is a relatively new research field and the following subsection (Sec. 2.3.1) reviews recent approaches that appeared in the literature in the last few years. Body pose dependency in the other hand, has been widely studied in the fields of example-based shape animation and modeling. These methods are reviewed in Sec. 2.3.2.

2.3.1. Image-Based Pose Synthesis

Little research has been done in pose-dependent image-based rendering techniques. One of the earliest works that approached the problem of learning articulated pose-dependent appearance was presented by Darrell [Dar98], who learned the silhouette appearance of an articulated arm as a function of its end point position using radial basis function networks. Later, methods that separated the scene into several independent rigid parts [CYJ02, XRS02] were proposed for textured non-rigid scenes. Both papers present an interpolation of an articulated human motion, where body parts are modeled as separate rigid objects. Other methods create novel poses from single images by deforming characters based on skeletons and meshes [HDK07, IMH05]. However, pure deformation of a single image cannot model pose-dependent appearance changes, such as changes in texture and shading. Moreover, if large deformations are required, distortions and deformation artifacts can occur. This has been approached by exploiting a larger number of prerecorded images [VGB06, XLS+11, HSR11b, HSR13]. These methods first learn and then search a database of pose-dependent appearance using a pose similarity measure. Vanaken et al. [VGB06] presented a method for image-based animation of articulated characters, which rearranges existing video frames to allow animation. However, only existing frames are exploited, and no deformation or animation takes place. Therefore, the captured poses have to be close to the output poses to render a smooth animation. The works presented in [SMH05, HHS09] exploit 3D models and multi-view video to create motion graphs of captured motions. New motions are synthesized by rearranging subsequences of the recorded motions. More recently, methods have been proposed that search a pre-recorded database of images for a texture to be mapped onto an animated 3D model of a human, based on pose information. This model can e.g. be a specific animated 3D model of the user created from a laser scan [XLS+11] or a visual hull created from a multi-view capture setup [HSR13, HSR11b]. The *best matching* texture is found based on pose-related information, e.g. skeleton joints [XLS+11] or silhouette information [HSR13, HSR11b], and the texture is finally warped to fit the silhouette of the 3D model. Similar to the pose-dependent image-based rendering approach presented in Chapter 4, in these approaches, the database search is performed based on pose-related information. However, in contrast to the method presented in this thesis, these methods do not extract pose-dependent characteristics from the database and can be seen as pose-dependent texture mapping: a single best fitting texture is searched for, i.e. image intensities are interpolated in a nearest-neighbor fashion in pose-space. Pose-dependent texture deformation is not learned a-priori but performed online during rendering based on silhouette fitting. This procedure requires a very dense sampling of the pose-space.

Regarding pose-dependent image-based rendering of clothes for augmented reality applications, such as virtual try-on in particular, only recently there have been some approaches towards this topic. Ehara and Saito [ES06] proposed a method to augment the image of a user with a shirt. From a database of images showing a person wearing a shirt with sparse markers, a mapping is established between

silhouette and texture deformation. During rendering, given an image of a person, the database is searched for a similar image based on silhouette information. A new arbitrary texture is deformed based on the stored texture deformation in the database and rendered into the original image of the user. The method was extended by Tanaka and Saito [TS09] to handle occlusions at the shirt boundary. Zhou et al. [ZSZ+12] recently proposed a system for image-based clothing animation in virtual fitting applications, which selects the best matching clothing image from a database of images based on 3D skeleton joint positions. The joint positions are then used to warp the selected image onto the pose in the input image. Similarly, Hauswiesner et al. [HSR13, HSR11a, HSR11b] follow a silhouette-based searching strategy to search a database of clothing images and find the most similar clothing image to an incoming user image. The database image is then warped onto the input silhouette image to texture a visual hull-based avatar of the user.

The most related approach to the one presented in this thesis is the work of Hauswiesner et al. [HSR13]. In this work, they propose an extension of their previous method [HSR11a, HSR11b], using different database images for different parts of the body. In contrast to the approach presented in this thesis, a single best fitting database image is exploited (per body part), and intensities are interpolated in a global nearest neighbor approach, requiring a very dense sampling of the pose-space. Furthermore, the synthesis, i.e. image warping, is not pose-dependent and not learned a-priori. In contrast to that, in the approach presented in this thesis, pose-dependent characteristics, such as texture deformation and shading properties, are implicitly extracted from the database images as warps between them, which are stored in the database and interpolated during rendering, allowing for a smooth and pose-dependent movement of fine details such as wrinkles and shading patterns.

2.3.2. Example-Based Animation and Modeling

While a pose-dependent representation is comparatively new to image-based rendering methods, interpolation in *pose-space* has been applied to geometry deformation in example-based animation and modeling techniques (e.g. to realistically model muscle bulging when an arm is bent), called *pose-space deformation* (PSD) [LCF00, SRC01, WSLG07, WHRO10, NVH+13]. These methods provide examples of pose-dependent shape and geometry of an animated object for a number of example poses. During animation, these examples guide the geometric deformation, which pure skinning methods like skeleton subspace deformation (SSD) cannot model. SSD is a widely used skinning technique, which is often used with linear blend skinning (LBS) (details can be found in [LCF00]). SSD with LBS transforms each vertex $\mathbf{v}$ of a (skinned) mesh by a weighted linear blend of bone transformations to

$$\tilde{\mathbf{v}} = \mathcal{SSD}(\mathbf{v}) = \sum_b w_b(\mathbf{v}) \cdot \delta\mathbf{W}_b \cdot \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} \ , \tag{2.1}$$

where $\tilde{\mathbf{v}}$ denotes the transformed vertex, $\delta\mathbf{W}_b$ is a homogeneous rigid transformation matrix of a skeleton bone b, and $w_b(\mathbf{v})$ is a *skinning weight*, associating the vertex $\mathbf{v}$

to the bone b. Contextual deformation, such as muscle bulging [NVH+13] or cloth wrinkling [WHRO10], is difficult to achieve with SSD, since the shapes are limited to a linear combination of transformations. In addition, SSD suffers from the collapsing joints defect [LCF00] where the skin near joints collapses for increasing bending or twisting angles. In pose-space deformation methods, these limitations are avoided by providing examples of shape for several body poses. Shapes of new poses are inter- or extrapolated from these examples using scattered data interpolation. Typically, the examples are modeled by the user, but examples from computationally demanding simulations [WHRO10], body scans [ACP02] or multi-view reconstruction [NVH+13] have also been used. In the latter cases, the scans have to be transformed to a parameterization that is consistent over all poses.

In general, shape deformations in PSD methods can be described by

$$\begin{aligned} \tilde{\mathbf{v}}(\mathbf{q}) &= \mathcal{SSD}\big(\mathbf{v} + \Delta\mathbf{v}(\mathbf{q})\big) \\ \text{or } \tilde{\mathbf{v}}(\mathbf{q}) &= \mathcal{SSD}(\mathbf{v}) + \Delta\mathbf{v}(\mathbf{q}) \ , \end{aligned} \tag{2.2}$$

where $\mathbf{q}$ represents an arbitrary pose parameterization, and $\mathcal{SSD}(\mathbf{v})$ is a function describing skeletal subspace deformation, as given in Equ. (2.1). $\Delta\mathbf{v}(\mathbf{q})$ denotes a vertex displacement as a function of pose, known for the provided examples. These examples are parameterized in *pose-space* by a suitable pose representation, e.g. a vector containing the joint angles of an underlying skeleton model. The pose-space is used as interpolation domain, in which the example poses are located at scattered positions. To generate the shape for a new point in this space, the vertex displacements are interpolated by scattered data interpolation methods, e.g. radial basis functions (RBF) [LCF00, SRC01] or k-nearest neighbors (kNN) interpolation [ACP02]. The example vertex displacements are either represented in the local coordinate frame of the nearest bone or calculated after all example poses have been transformed to the same base pose using SSD. The interpolation can either be performed on a per-example basis, assigning the same weight to all vertices of one example [LCF00, SRC01] or on a per vertex basis, assigning each vertex a different weight [KM04].

The pose representation $\mathbf{q}$ defines the high-dimensional interpolation domain, the *pose-space*, and is therefore crucial for the interpolation result. For articulated characters, the pose-space can be derived from the underlying skeleton, exploiting e.g. the angles of skeleton joints. Lewis et al. [LCF00] do not specify their pose-space parameterization in detail but mention that for an articulated body the space may consist of several subspaces for each joint (where e.g. the elbow subspace would have one degree of freedom and the shoulder subspace would have two or three degrees of freedom). Weber et al. [WSLG07] use log-quaternions of joint angles to represent poses. Wang et al. [WHRO10] propose to use an axis angle representation for joint rotations as a basis for the pose-space. Instead of storing displacements for key poses in a database, Kry et al. [KJP02] perform a principle component analysis of the deformations, which are then represented as reduced eigenbases, allowing for a more efficient interpolation.

Recently, example-based methods have been proposed specifically for clothing simulation to dress virtual characters [KV08, WHRO10, GRH+12]. One approach is to separate fine wrinkles from the coarse clothing shape and learn a pose-dependent wrinkling database from high accurate physical clothing simulation. During animation, a coarse physical-based simulation is used and high detailed wrinkles are synthesized from the database using example- or learning-based methods, e.g. pose-space deformation [KV08, WHRO10]. Guan et al. [GRH+12] extend these methods by addressing not only pose- but also body shape-dependency of wrinkling.´

2.4. Texture Replacement

Texture replacement or retexturing is the process of replacing a texture of a deformed or deforming surface in an image or video sequence with a synthetic one. To achieve a realistic retexturing result, various characteristics of the original surface or texture have to be maintained, such as geometric deformation, shading and visibility. This information can either be provided by a 3D model of the surface and the scene [SSK+05, WCF07, BPS+08], or it can directly be estimated from the image data (image-based retexturing) [FH04, PLF05a, WF06, SM06, PLF08, GSPJ08, BRB09, YS10, LSH+11]. In this case, texture deformation is often represented as a deformation field, whereas shading information is modeled as a shading map multiplied to the image intensities. Both the deformation field and and the shading map are then applied to a new synthetic texture, which is blended into the original image or video.

This section focuses on image-based retexturing methods. The following subsections distinguish between methods that exploit a given reference image of the undeformed texture (reference-based retexturing) and methods that estimate texture deformation and shading from a single image without such a reference (reference-free methods).

2.4.1. Reference-Based Retexturing

If a reference of the undeformed and uniformly lit texture is available, texture deformation and shading information can directly be estimated from an image [PLF05b, SM06, WF06, PLF08, GBBS10, BRB09]. This has been widely used to augment monocular video sequences with a new synthetic texture [PLF05b, SM06, PLF08, GBBS10, BRB09]. One approach is to use markers or color-coded patterns to estimate spatial texture deformation [BRB09, SM06]. These methods often establish a shading map by interpolating intensity information from regions between the markers. Other approaches restrict the surface to consist of a limited set of colors that can be easily classified [WF06]. The assumption of such a-priori knowledge, however, is problematic in many applications and limits the applicability for arbitrary video sequences. For arbitrary textures, deformation and shading estimation

are often performed separately, too. A spatial mapping or warp is first established between the two images, e.g. by feature-based [PLF05b, PLF08] or intensity-based methods [GBBS10], and intensities are compared based on this spatial mapping, e.g. the luminance ratio between the warped and the reference image is calculated to establish a shading map [PLF05b, PLF08]. If the registration is not accurate enough, however, this approach has problems at texture edges such that the old texture may still be visible under the synthetic one, requiring additional filtering of the shading map.

2.4.2. Reference-Free Texture Replacement

The retexturing methods described in the previous paragraph estimate surface deformation and shading properties in relation to a given reference, i.e. markers or an image of the undeformed and uniformly lit texture. However, such a reference is not always available and not easy to provide. Without a reference, texture deformation and shading have to be estimated directly from the input image. This is related to shape-from-shading and shape-from-texture approaches. These methods use shading or texture deformation as strong cues for depth to reconstruct the 3D structure of an object from a single image [ZTCS99, For02, LF06, HSE11b]. Consequently, current reference-free retexturing methods exploit shading [FH04, GSPJ08, YS10, LSH+11] or texture [LLH04] to estimate the texture deformation field and additional lighting conditions.

Shading-based methods extract shading maps directly from the input image and derive texture deformation from shading. This limits their applicability to untextured, diffuse surfaces illuminated by a single directional light source. Most methods estimate surface normals from the extracted shading maps and use these normals to calculate a texture deformation field [FH04, YS10]. Other methods assume that sharp intensity changes in the image represent creases and wrinkles. These approaches stretch and deform a mesh on the image based on gradient information to model texture distortion [GSPJ08, LSH+11]. In general, shading-based methods maintain the shading of the original image well, but the original (especially global) texture distortion is often destroyed, e.g. if an arm is bent.

Surprisingly, only few papers on retexturing exploit texture information to estimate deformation and shading. To be able to estimate texture deformation from a single image, methods have to be limited to specific types of texture. Many cloth patterns are of a regular type, i.e. the texture is generated by tiling the texture space with one or more repeating texture elements. Mathematically, such patterns can be defined by a pattern element (called texture element, texel or texton) and two smallest linearly independent generating vectors, describing the tiling pattern [GS86]. Textures that deviate geometrically and photometrically from a regular congruent tiling are often called near-regular textures (NRT) [LLH04, LTL05] (Fig. 2.1). Liu et al. [LLH04] were the first who analyzed near-regular textures as a topological

Figure 2.1.: Near-regular texture examples. Near-regular textures can be seen as geometrically and photometrically modified regular textures (images taken from the CMU NRT database [CMU]).

Type	0	I	II	III
Geometry	Regular	Regular	Irregular	Irregular
Color	Regular	Irregular	Regular	Irregular

Table 2.1.: A categorization of near-regular textures after [LLH04].

regular lattice structure under geometric and photometric deformation and categorized them based on their geometric and color irregularities (Tab. 2.1). A regular texture on a 3D surface projected into an image appears as an NRT due to variations in the viewing angle, lighting conditions and partial occlusions. Based on the categorization in Tab. 2.1, this is a near-regular texture of type NRT III - irregular geometry and irregular color. This type of near-regular textures provides strong cues for the perception of shape and deformation and is of wide interest in the computer vision and graphics community, e.g. for shape-from-texture applications [LLH04, For02, LF06, HSE11b].

Based on their definition of near-regular textures as warped regular textures, Liu et al. [LLH04] presented a user-assisted interactive method for near-regular texture analysis and retexturing. The basic idea is that for each near-regular texture there exists a topologically regular grid or lattice that describes a spatial deformation of that texture from a regular lattice. In their approach, the lattice generation needs interactive and highly accurate editing by the user. Once the lattice is defined by the user, the texture is straightened out, and a light map is extracted using the method of [TLR01]. Later, automatic lattice detection methods have been proposed for near-regular textures in real-world images to ease the lattice generation procedure, independent from retexturing purposes [HLEL06, PBCL09]. These methods search for visually similar and repeating interest points in the image and try to find a topologically consistent lattice between them. The lattice detection problem is formulated as a spatial multi-target tracking problem of repeating texture elements, in which each texel can undergo an affine transformation. This assumes that either a texel is indefinitely small such that the deformation can be described by locally rigid (in 3D) transformations of each texel or that the true 3D deformation is smooth. Complex deformations of texels are not modeled. The presented results show images of urban scenes and other regular textures on planes such that the deformation in the image mainly comes from projective transformations rather than true deformation in 3D. Applications like cloth retexturing has not been shown in

these works.

2.5. Virtual Try-on Systems

The main application of the visualization methods developed during the research for this thesis is the visualization of clothes in virtual try-on or virtual mirror applications (Fig. 1.1 on page 5). Such a system allows a user to virtually try-on specific products, e.g. shoes, jewelry or clothing. Typically, in real-time systems, the mirror is replaced by a large screen showing the mirrored image of the user, captured by one or more cameras, and the virtual product is rendered into the screen, following the user's movements. Systems are already available for rigid objects, such as shoes [EFR08] or glasses[1]. The visualization of clothes in such a system is much more challenging because of the complex drapery and movements of the garment. For a realistic impression, the virtual piece of clothing should follow the users movements smoothly and drape realistically according to the user's pose.

In recent years, there has been a huge increase in virtual try-on applications for clothes with various approaches. The first available systems installed in stores show static images of a piece of clothing on a screen, and the potential buyer has to pose in such a way that his/her image best fits to the static view of clothes on the screen[2]. Although the clothing visualization relies on real images of clothes, this results in a poorly immersive impression of a mirror because the clothes do not follow the user's motion. Other applications avoid real-time simulation of clothing by visualizing individualized clothes on a virtual counterpart (avatar) of the user, generated from a 3D Laser scan or designed by the user himself [CLSMT01, CSMT03, DTE+04, WKK+05, MTKV+11]. Within this approach there is a lack of identification of the customer with his/her virtual counterpart.
With recent development in video plus depth sensors and human body tracking methods (e.g. Microsoft Kinect [SFC+11]), recent methods render virtual and physically simulated clothes directly on the user's body on the screen. The user's shape and motion are roughly estimated and computer-generated clothes are physically simulated based on pose and shape information and mapped onto the user in the mirror image in real-time[3]. However, real-time performance is achieved at the expense of realism of the simulated virtual clothes, which have an artificial and unlifelike appearance. Thus, while real-time body pose estimation methods are available nowadays, realistic simulation of clothes is still the bottle-neck of real-time virtual try-on systems.

[1] Ray Ban: http://www.ray-ban.com/usa/science/virtual-mirror
Mister Spex: http://misterspex.de/brillen/brillenanprobe.html
[2] Try On Bathing Suit: http://www.tryonbathingsuit.com
Imagine That: http://www.imaginethattechnologies.com
Web Cam Social Shopper http://webcamsocialshopper.com
[3] Fitnect: http://www.fitnect.hu
AR-Door: http://ar-door.com

To sum up, existing approaches provide a real-time visualization but are quite limited in visual quality and plausibility. Others focus on high-quality cloth simulation and rendering but are far from working in real-time. Consequently, methods have recently been proposed in the computer vision and graphics community that rely on real images instead of physically simulated computer generated clothes. First approaches focused on texture overlay or retexturing of the user's real clothes [ES03, ES05, TS09], whereas recent methods search a database of clothing images to be mapped onto the moving body of a user in the mirror image [HSR13, HSR11a, HSR11b, ZSZ+12]. The database search is performed based on shape or pose parameters. The queried images are warped onto the shape of the user's body. These promising first approaches demonstrate that the use of real images instead of computer generated clothes results in a more photo-realistic appearance of the rendered clothes and hence a more immersive experience of a virtual try-on system.

3. Joint Spatial and Photometric Warp Optimization

The key idea of this dissertation is to exploit as much information from real images as possible for a realistic visualization of clothes. One key element in the following chapters will be the knowledge about *warps* between the images, i.e. transformation rules that map one image onto another. This chapter introduces a framework to estimate parameters of joint spatial and photometric warps, i.e. warps defined not only in the spatial domain but also in the intensity domain, thereby capturing meaningful information about texture deformation or shading.

An image can be represented as an array of pixels, each carrying an intensity or color value. A gray-value image can therefore be seen as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}$, assigning an intensity $\mathcal{I}(\mathbf{x})$ to each pixel $\mathbf{x} = [x\ y]^T$. Similarly, a color image can be seen as a function $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}^3$ mapping a pixel coordinate onto an RGB color vector $\mathcal{I}(\mathbf{x}) = \left[\mathcal{I}^R(\mathbf{x})\ \mathcal{I}^G(\mathbf{x})\ \mathcal{I}^B(\mathbf{x})\right]^T$. Usually, a warp refers to a spatial image transformation, changing the position of pixels. In the following, such a spatial warp will be denoted by

$$\tilde{\mathbf{x}} = \mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s) \ , \tag{3.1}$$

where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ represent the original and transformed pixel positions in the image, and $\boldsymbol{\theta}_s$ represents a vector of spatial warp parameters. A spatial image warp from a source image $\mathcal{I}_\mathcal{S}$ to a target image $\mathcal{I}_\mathcal{T}$ moves the pixels $\mathbf{x}$ to a new location such that the warped image $\mathcal{I}_\mathcal{S}(\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s))$ best resembles $\mathcal{I}_\mathcal{T}(\mathbf{x})$:

$$\mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)\big) \approx \mathcal{I}_\mathcal{T}(\mathbf{x}) \ . \tag{3.2}$$

However, differences between images cannot always be modeled by pure spatial warps e.g. because of varying lighting or shading patterns. Therefore, a photometric warp

$$\tilde{\mathcal{I}}(\mathbf{x}) = \mathcal{W}_p\big(\mathcal{I}(\mathbf{x}); \boldsymbol{\theta}_p\big) \tag{3.3}$$

with a photometric parameter vector $\boldsymbol{\theta}_p$ is introduced to additionally modify the intensities in the image such that the spatially and photometrically warped source image best resembles the target image:

$$\mathcal{W}_p\big(\mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)\big); \boldsymbol{\theta}_p\big) \approx \mathcal{I}_\mathcal{T}(\mathbf{x}) \ . \tag{3.4}$$

The knowledge of spatial and photometric warps between images yields information about texture deformation and shading (Chapter 5), (deformable) motion and

lighting changes in a video sequence (Sec. 3.3) or a large database of images (Chapter 4) as well as depth information in a multi-view setup (Chapter 4). Because photometric properties are important for the realistic appearance of synthetically generated images, the extracted information of the photometric warps is essential for the image-based rendering approaches presented in this thesis. In addition, the photometric warps make spatial image registration and tracking more robust against lighting changes than traditional image-based warp optimization methods (Sec. 3.3).

This chapter is organized as follows. Sec. 3.1 starts with a background on (spatial) image warp estimation based on the brightness constancy assumption and a mesh-based warp parametrization. The spatial image registration task is formulated as an optimization problem that solves for the warp parameters using non-linear optimization methods. Sec. 3.2 introduces an extension to joint spatial and photometric warp optimization for two different kinds of photometric warp models. An evaluation and analysis of the proposed warp optimization approach is presented in Sec. 3.3.

3.1. Image-Based Warp Optimization Revisited

Estimating a spatial warp between two images amounts to solving for the spatial warp parameter vector $\boldsymbol{\theta}_s$ that best transforms one image onto another (Equ. (3.2)). Generally, this can be formulated as a non-linear optimization problem minimizing a cost function that consists of two terms

$$\hat{\boldsymbol{\theta}}_s = \arg\min_{\boldsymbol{\theta}_s} \left(\mathcal{E}_D(\boldsymbol{\theta}_s) + \lambda^2 \mathcal{E}_S(\boldsymbol{\theta}_s) \right) \ , \tag{3.5}$$

where $\mathcal{E}_D(\boldsymbol{\theta}_s)$ is called the *data term* and $\mathcal{E}_S(\boldsymbol{\theta}_s)$ represents prior knowledge on the smoothness of the warp and is therefore often called the *smoothness term*. λ is a regularization parameter weighting the influence of this prior knowledge against fitting to the data term. $\boldsymbol{\theta}_s$ is the vector of spatial warp parameters (where the subscript $_s$ stands for *spatial* to later differentiate between spatial and photometric warp parameters). In *direct* or *image-based* optimization methods [HS81, IA99, BZ04, LY05], the data term is often formulated by exploiting the brightness constancy assumption [Hor86] (Equ. (3.24)), whose first Taylor expansion is also called the *optical flow equation*. Both the data as well as the smoothness term will be derived in this section and can be formulated using norm-like robust functions. The resulting cost function can be minimized using Quasi-Newton approaches, like Gauss-Newton (GN) or Levenberg-Marquardt (LM), iteratively solving for a parameter update $\delta\hat{\boldsymbol{\theta}}_s$ and updating the parameter vector by $\hat{\boldsymbol{\theta}}_s \leftarrow \hat{\boldsymbol{\theta}}_s + \delta\hat{\boldsymbol{\theta}}_s$.

This section is organized as follows. Sec. 3.1.1 starts with a brief background on the exploited non-linear optimization methods, before Sec. 3.1.2 focuses on spatial warp optimization, including the derivation of the data term from the brightness constancy assumption (Sec. 3.1.2.1), the parametrization of warps by mesh-based models (Sec. 3.1.2.2), and a formulation of the smoothness term based on these models (Sec. 3.1.2.3).

3.1.1. Non-Linear Optimization Techniques

This section shortly recapitulates non-linear optimization techniques exploited in the image-based warp estimation framework to minimize a cost function of a form as given in Equ. (3.5). Details are given in [MNT04, NW00].

Gauss-Newton Optimization. A general formulation of a non-linear least-squares problem with m equations and n unknowns $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \dots \ \theta_n]^T$ is given by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) \ , \tag{3.6}$$

where $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}$ is a cost function of the form

$$\mathcal{E}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{m} r_i(\boldsymbol{\theta})^2 = \frac{1}{2} \left\| \mathbf{r}(\boldsymbol{\theta}) \right\|^2 \ , \tag{3.7}$$

and $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$ is called a *residual vector* $\mathbf{r}(\boldsymbol{\theta}) = [r_1(\boldsymbol{\theta}) \ \dots \ r_m(\boldsymbol{\theta})]^T$. The gradient of $\mathcal{E}(\boldsymbol{\theta})$ is given by

$$\mathbf{g}_{\mathcal{E}}(\boldsymbol{\theta}) = \nabla \mathcal{E}(\boldsymbol{\theta}) = \mathbf{J}_{\mathbf{r}}(\boldsymbol{\theta})^T \cdot \mathbf{r}(\boldsymbol{\theta}) \ , \tag{3.8}$$

where $\mathbf{J}_{\mathbf{r}}(\boldsymbol{\theta})$ denotes the $m \times n$ Jacobian matrix of $\mathbf{r}(\boldsymbol{\theta})$. The Hessian of $\mathcal{E}(\boldsymbol{\theta})$ is given by

$$\begin{aligned} \mathbf{H}_{\mathcal{E}}(\boldsymbol{\theta}) = \nabla^2 \mathcal{E}(\boldsymbol{\theta}) &= \mathbf{J}_{\mathbf{r}}^T(\boldsymbol{\theta}) \mathbf{J}_{\mathbf{r}}(\boldsymbol{\theta}) + \sum_{i=1}^{m} r_i(\boldsymbol{\theta}) \cdot \nabla^2 r_i(\boldsymbol{\theta}) \\ &\approx \mathbf{J}_{\mathbf{r}}^T(\boldsymbol{\theta}) \mathbf{J}_{\mathbf{r}}(\boldsymbol{\theta}) \ . \end{aligned} \tag{3.9}$$

The Gauss-Newton (GN) method is based on a linear approximation of $\mathbf{r}(\boldsymbol{\theta})$ in the neighborhood of $\hat{\boldsymbol{\theta}}$ using a first order Taylor expansion:

$$\mathbf{r}(\hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) \approx \mathbf{r}(\hat{\boldsymbol{\theta}}) + \mathbf{J}_{\mathbf{r}}(\hat{\boldsymbol{\theta}}) \cdot \delta\boldsymbol{\theta} \ . \tag{3.10}$$

This leads to the following approximation of the cost function $\mathcal{E}(\boldsymbol{\theta})$:

$$\mathcal{E}(\hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) \approx \frac{1}{2} \left\| \mathbf{r}(\hat{\boldsymbol{\theta}}) + \mathbf{J}_{\mathbf{r}}(\hat{\boldsymbol{\theta}}) \cdot \delta\boldsymbol{\theta} \right\|^2 \ . \tag{3.11}$$

In each iteration step, the minimization problem is thus reduced to a linear least-squares problem, which is solved by the *normal equations*

$$\delta\boldsymbol{\theta}_{\text{GN}} = - \left(\mathbf{J}_{\mathbf{r}}^T \mathbf{J}_{\mathbf{r}} \right)^{-1} \mathbf{J}_{\mathbf{r}}^T \mathbf{r} \approx -\mathbf{H}_{\mathcal{E}}^{-1} \mathbf{g}_{\mathcal{E}} \ , \tag{3.12}$$

with $\mathbf{J}_{\mathbf{r}} = \mathbf{J}_{\mathbf{r}}(\hat{\boldsymbol{\theta}})$ and $\mathbf{r} = \mathbf{r}(\hat{\boldsymbol{\theta}})$, both evaluated at the current parameter estimation $\hat{\boldsymbol{\theta}}$. In each iteration step, $\hat{\boldsymbol{\theta}}$ is updated by $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}$.

Levenberg-Marquardt Optimization. The Gauss-Newton method does not guarantee convergence, because the parameter update vector $\delta\boldsymbol{\theta}$ can point downhill

but might be too long. Therefore, Levenberg [Lev44] and later Marquardt [Mar63] proposed to use a *damped* Gauss-Newton method to solve an error function of the form given in Equ. (3.7). This method calculates the parameter update by

$$\delta\boldsymbol{\theta}_{\text{L}} = -(\mathbf{J_r}^T\mathbf{J_r} + \alpha\mathbf{I}_n)^{-1}\mathbf{J_r}^T\mathbf{r} \ , \tag{3.13}$$

with $\mathbf{J_r} = \mathbf{J_r}(\hat{\boldsymbol{\theta}})$ and $\mathbf{r} = \mathbf{r}(\hat{\boldsymbol{\theta}})$. α is a *damping factor*, which interpolates the direction between the Gauss-Newton direction ($\alpha \to 0$) and a gradient descent step ($\alpha \to \infty$) [MNT04, NW00]. Often, α is chosen heuristically, e.g. if the error decreases, α is updated by $\alpha \leftarrow 0.1\alpha$ and if the error increases by $\alpha \leftarrow 10\alpha$. Equ. (3.13) minimizes

$$\frac{1}{2}\left\|\mathbf{r}(\hat{\boldsymbol{\theta}}) + \mathbf{J_r}(\hat{\boldsymbol{\theta}})\delta\boldsymbol{\theta}\right\|^2 + \frac{1}{2}\alpha^2\left\|\delta\boldsymbol{\theta}\right\|^2$$

in each iteration step, whereas the Gauss-Newton step in Equ. (3.12) minimizes

$$\frac{1}{2}\left\|\mathbf{r}(\hat{\boldsymbol{\theta}}) + \mathbf{J_r}(\hat{\boldsymbol{\theta}})\delta\boldsymbol{\theta}\right\|^2$$

such that the damping method can be seen as a Tikhonov regularization (page 28) penalizing large steps in each iteration [TGSY95, ABT05].

Marquardt [Mar63] later improved this method with an incorporation of estimated local curvature by

$$\delta\boldsymbol{\theta}_{\text{LM}} = -\left(\mathbf{J_r}^T\mathbf{J_r} + \alpha\cdot\operatorname{diag}(\mathbf{J_r}^T\mathbf{J_r})\right)^{-1}\mathbf{J_r}^T\mathbf{r} \ , \tag{3.14}$$

where $\operatorname{diag}(\mathbf{J_r}^T\mathbf{J_r})$ denotes a diagonal matrix consisting of the diagonal elements of $\mathbf{J_r}^T\mathbf{J_r}$.

Robust Optimization. Equ. (3.7) formulates the least-squares optimization problem. A *generalized* optimization problem is given by minimizing

$$\mathcal{E}(\boldsymbol{\theta}) = \sum_{i=1}^{m}\psi\big(r_i(\boldsymbol{\theta})\big) \ . \tag{3.15}$$

Usually, $\psi(r_i(\boldsymbol{\theta}))$ is a *robust estimator* [Wed74, MN89], i.e. a symmetric, positive-definite function with a unique minimum at zero, and less increasing than square. The general idea is to give large residuals less weight to make the optimization more robust against outliers in the data (Fig. 3.1). The minimization can be computed by an iteratively reweighted least-squares method, which is a Gauss-Newton type method for minimizing a sum of norm-like functions of the residuals [Hub81, AJ09]. With an *influence function*

$$\vartheta(r_i) = \frac{\partial\psi(r_i)}{\partial r_i} \tag{3.16}$$

and a *weight function*

$$w(r_i) = \frac{\vartheta(r_i)}{r_i} \ , \tag{3.17}$$

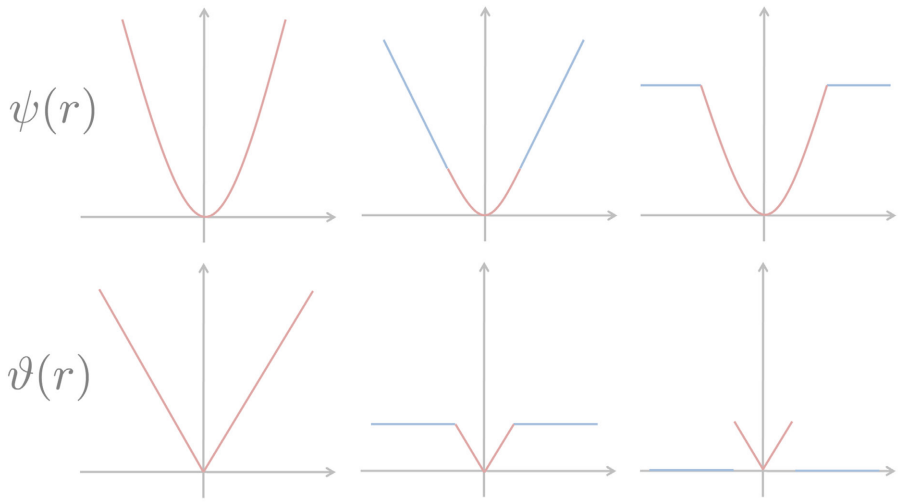


Figure 3.1.: Estimator kernel $\psi(r)$ and influence function $\vartheta(r)$ for different robust estimators. From left to right: least-squares function; Huber function; Talwar function. For the non-robust least-squares estimator, the influence function of a datum on the estimate increases linearly with the size of its error.

a weighting matrix $\mathbf{W}$ can be defined, and the parameter update is defined by

$$\delta\boldsymbol{\theta}_{\mathrm{r}} = -\left(\mathbf{J}_{\mathbf{r}}^{T}\mathbf{W}\mathbf{J}_{\mathbf{r}}\right)^{-1}\mathbf{J}_{\mathbf{r}}^{T}\mathbf{W}\mathbf{r}\ , \tag{3.18}$$

with $\mathbf{J}_{\mathbf{r}} = \mathbf{J}_{\mathbf{r}}(\hat{\boldsymbol{\theta}})$, $\mathbf{r} = \mathbf{r}(\hat{\boldsymbol{\theta}})$, and $\mathbf{W} = \mathrm{diag}\big(w\big(\mathbf{r}(\hat{\boldsymbol{\theta}})\big)\big)$. Parameter updates for a robust Levenberg-Marquardt method are equivalent.

The influence function $\vartheta(r_i)$ measures the influence of a datum on the value of the parameter estimate (Fig. 3.1). One efficient robust estimator is the Huber function [Hub81]

$$\psi_{\mathrm{H}}(r_i) = \begin{cases} \frac{1}{2}r_i^2 & \text{if } |r_i| \leq \kappa_{\mathrm{H}} \\ \kappa_{\mathrm{H}}\,|r_i| - \frac{1}{2}\kappa_{\mathrm{H}}^2 & \text{else} \end{cases}\ , \tag{3.19}$$

which is a parabola in the vicinity of zero and increases linearly at a given level above a threshold κ_{H} (Fig. 3.1). The weight function for the Huber kernel is given by

$$w(r_i) = \begin{cases} 1 & \text{if } |r_i| \leq \kappa_{\mathrm{H}} \\ \frac{\kappa_{\mathrm{H}}}{|r_i|} & \text{else} \end{cases}\ . \tag{3.20}$$

The Huber function does not reject outliers, i.e. residuals with $|r_i| > \kappa_{\mathrm{H}}$, completely, in contrast to e.g. the Talwar function (Fig. 3.1)

$$\psi_{\mathrm{T}}(r_i) = \begin{cases} \frac{1}{2}r_i^2 & \text{if } |r_i| \leq \kappa_{\mathrm{T}} \\ \frac{1}{2}\kappa_{\mathrm{T}}^2 & \text{else} \end{cases}\ , \tag{3.21}$$

but instead gives these outliers less weight than in the least-squares approach [Hub81].

Tikhonov Regularization. Tikhonov regularization is a commonly used method for the regularization of ill-posed problems. To give preference to a particular solution with desirable properties, a regularization term is included in the optimization problem, yielding a damped least-squares problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(\mathcal{E}(\boldsymbol{\theta}) + \lambda^2 \left\| \Gamma(\boldsymbol{\theta}) \right\|^2 \right) \ . \tag{3.22}$$

$\Gamma(\boldsymbol{\theta})$ is called a *Tikhonov regularizor*, and λ is a regularization parameter controlling the influence of the side constraint on the estimate. A dominating approach of regularization is to require that the L^2-norm or an appropriate seminorm of the solution is small. An initial estimate $\hat{\boldsymbol{\theta}}$ of the solution may be included in the side constraint. In this case, the Tikhonov regularizor is given by:

$$\Gamma(\boldsymbol{\theta}) = \boldsymbol{\Gamma} \cdot (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \ .$$

In the simplest case (zeroth-order Tikhonov regularization), $\boldsymbol{\Gamma}$ is the identity matrix $\boldsymbol{\Gamma} = \mathbf{I}$. Other possible Tikhonov regularizations include first or second order Tikhonov regularizations [TGSY95, ABT05].

3.1.2. Spatial Warp Optimization

This section formulates the data term $\mathcal{E}_D(\boldsymbol{\theta}_s)$ (Sec. 3.1.2.1) as well as the smoothness term $\mathcal{E}_S(\boldsymbol{\theta}_s)$ (Sec. 3.1.2.3) of the cost function given for spatial image warp estimation:

$$\mathcal{E}(\boldsymbol{\theta}_s) = \mathcal{E}_D(\boldsymbol{\theta}_s) + \lambda^2 \mathcal{E}_S(\boldsymbol{\theta}_s) \ . \tag{3.23}$$

Mesh-based warp models, which will be exploited in this dissertation, are introduced in Sec. 3.1.2.2. Generally, in the presented framework the optimization method is independent from the specific warp model, and any other warp model can be incorporated.

3.1.2.1. Data Term: Brightness Constancy Assumption

Image-based warp estimation methods are usually based on the brightness constancy assumption, which assumes that an image pixel, representing an object point, does not change its brightness value but is allowed to change its position in the image. Hence, differences between two images $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$ are caused by spatial deformation only and can therefore be fully explained by a spatial warp function $\tilde{\mathbf{x}} = \mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)$ that moves a pixel to another location [Hor86]:

$$\mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)\big) = \mathcal{I}_\mathcal{T}(\mathbf{x}) \ . \tag{3.24}$$

In general, the spatial warp function depends on the pixel location $\mathbf{x}$ and is parameterized by a $N_s \times 1$ parameter vector $\boldsymbol{\theta}_s$, where N_s denotes the degrees of freedom of the spatial warp. Possible warp functions range from global translation with $N_s = 2$

over affine transformations with $N_s = 6$ to more complex models like mesh-based warps with $N_s = 2K$, where K is the number of mesh vertices, or a dense pixel displacement field with two parameters per pixel, i.e. $N_s = 2P$, where P is the number of pixels. Sec. 3.1.2.2 will introduce mesh-based warp parameterizations, exploited in this thesis.

A cost function over the warp parameters $\boldsymbol{\theta}_s$ can be defined by the sum of all pixel-wise errors between the warped source image and the target image in the region of interest $\mathcal{R}$ as

$$\begin{aligned} \mathcal{E}_D(\boldsymbol{\theta}_s) &= \sum_{\mathbf{x}_i \in \mathcal{R}} \psi\big(r_i(\boldsymbol{\theta}_s)\big) \\ r_i(\boldsymbol{\theta}_s)) &= \mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x}_i;\boldsymbol{\theta}_s)\big) - \mathcal{I}_\mathcal{T}\left(\mathbf{x}_i\right) \ , \end{aligned} \tag{3.25}$$

where ψ denotes any norm-like function, e.g. a least-squares function or a robust estimator like the Huber function (Equ. (3.19)). Minimizing $\mathcal{E}_D(\boldsymbol{\theta}_s)$ is a non-linear optimization problem, which can be solved using Quasi-Newton methods, like robust Gauss-Newton or Levenberg-Marquardt (Sec. 3.1.1). The Jacobian $\mathbf{J}_\mathbf{r}(\boldsymbol{\theta}_s)$ of the residuals in Equ. (3.25) is a $P \times N_s$ matrix with the following i^{th} row for pixel $\mathbf{x}_i \in \mathcal{R}, i = 1 \dots P$:

$$\frac{\partial r_i(\hat{\boldsymbol{\theta}}_s)}{\partial \boldsymbol{\theta}_s} = \nabla \mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x}_i;\hat{\boldsymbol{\theta}}_s)\big)^T \cdot \mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i;\hat{\boldsymbol{\theta}}_s) \ . \tag{3.26}$$

Here, $\mathcal{R}$ denotes the region of interest, and $\nabla\mathcal{I}_\mathcal{S}$ denotes the spatial derivatives of the image $\mathcal{I}_\mathcal{S}$[1]. $\mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i;\hat{\boldsymbol{\theta}}_s)$ is the Jacobian of the spatial warp function at pixel $\mathbf{x}_i$ evaluated at the current parameter estimates $\hat{\boldsymbol{\theta}}_s$. It depends on the warp parameterization and generally is a $2 \times N_s$ matrix. For mesh-based models used in this dissertation, it is derived in Sec. 3.1.2.2 (Equ. (3.29)).

3.1.2.2. Mesh-Based Warp Models

The most general formulation of a spatial image warp is a 2-dimensional (2D) pixel displacement field $\boldsymbol{\mathcal{D}}(\mathbf{x};\boldsymbol{\theta}_s)$ defined at each image pixel $\mathbf{x}_i$ and added to the pixel coordinates, parameterized by a $N_s \times 1$ parameter vector $\boldsymbol{\theta}_s$. This can be expressed by

$$\tilde{\mathbf{x}}_i = \mathcal{W}_s(\mathbf{x}_i;\boldsymbol{\theta}_s) = \mathbf{x}_i + \boldsymbol{\mathcal{D}}(\mathbf{x}_i;\boldsymbol{\theta}_s) \ , \tag{3.27}$$

where $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_i$ denote the original and transformed pixel positions. Based on the given image registration problem (type of motion, required accuracy of the deformation field, real-time constraints etc.), different types of displacement field parametrization can be chosen. If the parameterization is linear in the parameter vector $\boldsymbol{\theta}_s$, Equ. (3.27) can be written as

$$\mathcal{W}_s\left(\mathbf{x}_i;\boldsymbol{\theta}_s\right) = \mathbf{x}_i + \mathbf{B}_s(\mathbf{x}_i) \cdot \boldsymbol{\theta}_s \ , \tag{3.28}$$

[1] In practice, temporal averaging of the gradients can increase accuracy [HS81, SRB10].

where $\mathbf{B}_s(\mathbf{x}_i)$ is a $2 \times N_s$ matrix and will be called *warp parameterization matrix* for $\mathcal{W}_s$ in the following. This matrix is different for each pixel and depends on the parameterization type of the displacement field. For dense and mesh-based parameterizations, it will be defined below in Equ. (3.32)-(3.33) and (3.39)-(3.40). The warp formulation in Equ. (3.28) allows a straightforward derivation of the warp Jacobian, which will be required for the optimization:

$$\mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i; \boldsymbol{\theta}_s) = \mathbf{B}_s(\mathbf{x}_i) \ . \tag{3.29}$$

The simplest parameterization of the displacement field is a dense pixel grid with one displacement vector for each pixel. This *dense model* holds two parameters per pixel, i.e. the pixel displacements in x- and y-direction,

$$\begin{aligned} \Delta\mathbf{x} &= [\Delta x_1 \dots \Delta x_P]^T \\ \Delta\mathbf{y} &= [\Delta y_1 \dots \Delta y_P]^T \ , \end{aligned} \tag{3.30}$$

where P is the number of pixels in the region of interest. The spatial warp parameter vector ($N_s = 2P$) can then be defined by concatenating the pixel displacements in x-and y-direction to

$$\boldsymbol{\theta}_s = \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{bmatrix} \ , \tag{3.31}$$

and the warp parameterization matrix $\mathbf{B}_s(\mathbf{x}_i)$ is defined as follows. Let $\mathbf{b}(\mathbf{x}_i)$ be a pixel-dependent $1 \times P$ row vector with entries

$$b_j(\mathbf{x}_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \ , \tag{3.32}$$

where $b_j(\mathbf{x}_i)$ denotes the j^{th} entry in $\mathbf{b}(\mathbf{x}_i)$. The warp parametrization matrix $\mathbf{B}_s(\mathbf{x}_i)$ for the dense model at pixel $\mathbf{x}_i$ is then given by

$$\mathbf{B}_s(\mathbf{x}_i) = \begin{bmatrix} \mathbf{b}(\mathbf{x}_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{b}(\mathbf{x}_i) \end{bmatrix} \ . \tag{3.33}$$

The dense model is equivalent to the classical optical flow. Because the dense model holds two parameters per pixel, the normal equations in the optimization are rank deficient and additional (external) regularization is needed.

To reduce the number of parameters (internal regularization) and to introduce prior knowledge on the smoothness of the deformation field, 2D *mesh-based warp models* [PLF08, GBBS10] are used in this thesis. Let $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ denote a mesh consisting of a set of K vertices $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_K]^T$, with vertex positions $\mathbf{v}_k = [u \ v]^T$ in the image, and a set of faces $\mathcal{F}$, e.g. triangles (Fig. 3.2). A spatial image warp between two images $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$ can then be fully defined by a mesh $\mathcal{M}_\mathcal{S} : \{\mathbf{V}, \mathcal{F}\}$ on the source image $\mathcal{I}_\mathcal{S}$ and additional vertex displacements $\Delta\mathbf{V} = [\Delta\mathbf{v}_1 \dots \Delta\mathbf{v}_K]^T$ such that corresponding vertex positions are defined on the target image by $\mathcal{M}_\mathcal{T} : \{\mathbf{V} + \Delta\mathbf{V}, \mathcal{F}\}$:

$$\mathcal{W}_s : \{\mathbf{V}, \mathcal{F}, \Delta\mathbf{V}\} \ . \tag{3.34}$$

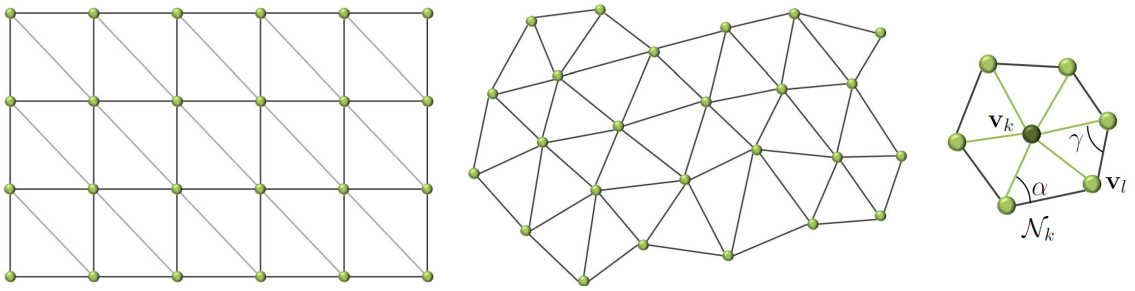


Figure 3.2.: Left: regular mesh structure. Center: non-regular mesh structure, e.g. generated with the method of [She96] where constraints, such as fixed vertex positions or mesh borders, can be provided as input for the triangulation. Right: neighborhood $\mathcal{N}_k$ of a vertex $\mathbf{v}_k$. α and γ define the angles used in the cotangent Laplacian (Equ. (3.45)).

This yields two parameters per vertex, i.e. its displacements in x- and y-direction,

$$\Delta\mathbf{V} = [\Delta\mathbf{v}_1 \dots \Delta\mathbf{v}_K]^T = [\Delta\mathbf{u}\ \Delta\mathbf{v}] \ , \tag{3.35}$$

with

$$\begin{aligned} \Delta\mathbf{u} &= [\Delta u_1 \dots \Delta u_K]^T \\ \Delta\mathbf{v} &= [\Delta v_1 \dots \Delta v_K]^T \ . \end{aligned} \tag{3.36}$$

The $N_s \times 1$ parameter vector $\boldsymbol{\theta}_s$ for the mesh-based warp can then be formulated similarly to Equ. (3.31) by concatenating the vertex displacements:

$$\boldsymbol{\theta}_s = \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{v} \end{bmatrix} \ . \tag{3.37}$$

A triangular mesh-based model can be parameterized using barycentric coordinates for interpolation between the vertex positions. If $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ denote the vertices of the mesh triangle surrounding pixel $\mathbf{x}_i$ and $\beta_1, \beta_2, \beta_3$ are the corresponding barycentric coordinates, a warp with piecewise affine interpolation between the respective three surrounding vertex positions keeps the barycentric coordinates constant. The spatial warp $\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)$ can then be parameterized by

$$\begin{aligned} \mathcal{W}_s\left(\mathbf{x}_i; \boldsymbol{\theta}_s\right) &= \mathbf{x}_i + \boldsymbol{\mathcal{D}}\left(\mathbf{x}_i; \boldsymbol{\theta}_s\right) \\ &= \mathbf{x}_i + \sum_{l=1}^{3} \beta_l \Delta\mathbf{v}_l \ . \end{aligned} \tag{3.38}$$

This can be formulated in matrix notation according to Equ. (3.28) by

$$\mathcal{W}_s\left(\mathbf{x}_i; \boldsymbol{\theta}_s\right) = \mathbf{x}_i + \mathbf{B}_s(\mathbf{x}_i) \cdot \boldsymbol{\theta}_s \ ,$$

and the warp parameterization matrix $\mathbf{B}_s(\mathbf{x}_i)$ can be defined similarly to the dense warp model as in Equ. (3.33) by

$$\mathbf{B}_s(\mathbf{x}_i) = \begin{bmatrix} \mathbf{b}(\mathbf{x}_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{b}(\mathbf{x}_i) \end{bmatrix} \ . \tag{3.39}$$

Now, $\mathbf{b}(\mathbf{x}_i)$ is a $1 \times K$ row vector for pixel $\mathbf{x}_i$ with entries

$$b_j(\mathbf{x}_i) = \begin{cases} \beta_l & \text{if } \mathbf{v}_j \text{ is the } l^{th} \text{ vertex in the triangle surrounding } \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} , \tag{3.40}$$

where $b_j(\mathbf{x}_i)$ denotes the j^{th} entry in $\mathbf{b}(\mathbf{x}_i)$.

3.1.2.3. Smoothness Term: Laplacian Mesh Smoothness

It is a reasonable assumption for most deformable image registration tasks that the deformation field is smooth. This assumption is often expressed by a smoothness constraint $\mathcal{E}_S(\boldsymbol{\theta}_s)$ in the cost function (Equ. (3.23)). This constraint associates a cost with undesired, non-smooth model states and regularizes the warp parameters with a suitably chosen Tikhonov regularizor:

$$\begin{aligned} \mathcal{E}_S(\boldsymbol{\theta}_s) &= \sum_k \psi\big(s_k(\boldsymbol{\theta}_s)\big) \\ \mathbf{s}(\boldsymbol{\theta}_s) &= \left[s_1(\boldsymbol{\theta}_s) \ \dots \ s_{N_s}(\boldsymbol{\theta}_s)\right]^T = \boldsymbol{\Gamma} \cdot \boldsymbol{\theta}_s \ . \end{aligned} \tag{3.41}$$

ψ denotes a robust estimator or the least-squares estimator. The Jacobian of the smoothness term is given by:

$$\mathbf{J}_\mathbf{s}(\boldsymbol{\theta}_s) = \boldsymbol{\Gamma} \ . \tag{3.42}$$

Often, smoothness is associated with a vanishing second derivative of a function. Hence, to force a warp to be smooth, the Tikhonov matrix is often chosen to approximate the second-order derivatives of the parameters. For mesh-based models (Sec. 3.1.2.2), the mesh *Laplacian* provides such a discrete approximation of a second derivative [WMKG07], as detailed in the following.

Laplacian Smoothness Term. Let f be any function defined at the vertices of a mesh $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ with K vertices. The function values can be e.g. vertex coordinates, normals, displacements or photometric parameters. The discrete Laplace operator, i.e. a second derivative operator, at vertex $\mathbf{v}_k$ is often defined as [Tau95, NISA06, BKP+10]

$$\Delta f(\mathbf{v}_k) = \sum_{n \in \mathcal{N}_k} w_{kn} \cdot f(\mathbf{v}_n) - f(\mathbf{v}_k) \ , \tag{3.43}$$

where $\mathcal{N}_k$ denotes the neighborhood of vertex $\mathbf{v}_k$, i.e. the set of vertices $\mathbf{v}_k$ is connected to (Fig. 3.2), and w_{kn} are positive numbers with $\sum_{n \in \mathcal{N}_k} w_{kn} = 1$. The weights can be chosen in various ways, accounting for the topology of the mesh or the neighborhood structure of the vertices. A simple choice is to use

$$w_{kn} = \frac{1}{|\mathcal{N}_k|} \ , \tag{3.44}$$

where $|\mathcal{N}_k|$ denotes the number of vertices in $\mathcal{N}_k$ (Fig. 3.2). A Laplacian with this weighting scheme is often called a *uniform* Laplacian. It solely depends on

the connectivity (topology) of the mesh and not on the spatial distribution of the vertices. A more general weighting scheme uses a function $\Phi(\mathbf{v}_k, \mathbf{v}_l)$ defined on the edges of neighboring vertices [Tau95, NISA06, BKP+10]:

$$w_{kl} = \frac{\Phi(\mathbf{v}_k, \mathbf{v}_l)}{\sum_{n \in \mathcal{N}_k} \Phi(\mathbf{v}_k, \mathbf{v}_n)} \ .$$

For example, $\Phi(\mathbf{v}_k, \mathbf{v}_n)$ can be the surface area of the two triangles that share the edge, or the length of the edge. A popular choice defines the *cotangent* Laplacian [NISA06, BKP+10] with

$$\Phi(\mathbf{v}_k, \mathbf{v}_l) = \cot\alpha + \cot\gamma \ , \tag{3.45}$$

where α and γ are the angles that lie opposite to the edge between $\mathbf{v}_k$ and $\mathbf{v}_l$ (Fig. 3.2).

If $\mathbf{f}$ defines the concatenated vector of function values associated with all vertices $\mathbf{f} = \begin{bmatrix} f_1 & \dots & f_K \end{bmatrix}^T$, the Laplacian of the entire mesh can be obtained by a matrix multiplication with the Laplace matrix $\mathbf{L}$:

$$\Delta\mathbf{f} = \mathbf{L} \cdot \mathbf{f} \ .$$

The Laplace matrix $\mathbf{L}$ is a $K \times K$ matrix with elements

$$l_{kn} = \begin{cases} -1 & k = n \\ w_{kn} & \mathbf{v}_n \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases} \ . \tag{3.46}$$

The Laplace matrix is exploited to formulate a smoothness term for the mesh parameters as defined in Equ. (3.41). As the parameter vector $\boldsymbol{\theta}_s = \begin{bmatrix} \Delta\mathbf{u}^T & \Delta\mathbf{v}^T \end{bmatrix}^T$ of a mesh-based spatial warp consists of two attributes per vertex, the Tikhonov matrix is composed of two Laplacians $\mathbf{L}$, one for each parameter set:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix} \ . \tag{3.47}$$

By using a Tikhonov matrix as given in Equ. (3.47) in the optimization scheme, $\mathcal{E}_S(\boldsymbol{\theta}_s)$ penalizes the discrete second derivative of the parameter distributions, i.e. it smooths the displacement field over the mesh [Tau95, WMKG07]. It dominates in textureless areas as well as for vertices detected as outliers when using e.g. the Huber function in the data term for robust estimation. If ψ in Equ. (3.41) is the Huber kernel (Equ. (3.19)), small values of $\mathbf{s}$ are penalized quadratically, while larger values are penalized linearly, allowing for discontinuities in the warp smoothness to some extend.

Laplacians and Mesh Boundaries. The distribution of the function values over the mesh is perfectly smooth in the aforementioned sense if

$$\|\Delta\mathbf{f}(\mathbf{v})\|^2 = \|\mathbf{L} \cdot \mathbf{f}(\mathbf{v})\|^2 = 0 \ ,$$

i.e. the discrete second derivative vanishes. The Laplacian defined in Equ. (3.43) does not treat the mesh borders different from the vertices inside the mesh. Therefore, smoothing with these Laplacians is known to shrink from the mesh border to the center when e.g. directly applied to the vertex coordinates [GP10]. This means that special treatment at the mesh borders is necessary. One solution is to take only those neighbors into account that have a *directional counterpart* or to delete all rows in the Laplace matrix with a row sum unequal zero.

3.2. Joint Spatial and Photometric Warp Optimization

The brightness constancy assumption in Equ. (3.24) assumes that an image pixel does not change its brightness and, hence, differences between two images $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$ can be fully explained by moving the pixels to another location. However, this assumption is not necessarily valid for natural scenes. Usually, two images in a video sequence or a multi-view setup do not only differ spatially, but also the intensity of a scene point can vary because of changes in the scene lighting, shading properties etc. If not handled correctly, varying lighting can impair spatial registration based on the brightness constancy assumption. One approach to make the warp estimation more robust against lighting changes is to formulate a cost function based on gradient values instead of intensity values [BBPW04] or to use lighting adaptation after each optimization step [SHE11a]. However, for the targeted purposes not only robustness against lighting changes is needed but also actual *retrieval* or *extraction* of shading or lighting differences between images in a photometric warp, e.g. to generate a shading map for realistic retexturing (Chapter 5). In this section, this is addressed by relaxing the brightness constancy assumption and incorporating an additional photometric warp between the two images, i.e. a warp in the intensity domain [HE09a] (Sec. 3.2.1). For color images, a color warp with a reduced parameter set will be introduced in Sec. 3.2.2. This warp model assumes the color (gain) between the two images to be a global parameter, but the intensity can vary locally [HE09b, HSE10]. Sec. 3.2.3 summarizes the optimization framework for joint spatial and photometric warp estimation, before Sec. 3.2.4 details relevant implementation details.

3.2.1. Incorporating a Photometric Warp

In general, a photometric warp, altering the intensities of an image, can be denoted by $\tilde{\mathcal{I}}(\mathbf{x}) = \mathcal{W}_p\big(\mathcal{I}(\mathbf{x}); \boldsymbol{\theta}_p\big)$, where $\boldsymbol{\theta}_p$ contains the photometric warp parameters. In this thesis, a photometric warp is defined as a multiplicative intensity scale field or *shading map* (Fig. 3.3), depending on the pixel position:

$$\mathcal{W}_p\big(\mathcal{I}(\mathbf{x}); \boldsymbol{\theta}_p\big) \equiv \mathcal{W}_p(\mathbf{x}; \boldsymbol{\theta}_p) \cdot \mathcal{I}(\mathbf{x}) \ .$$

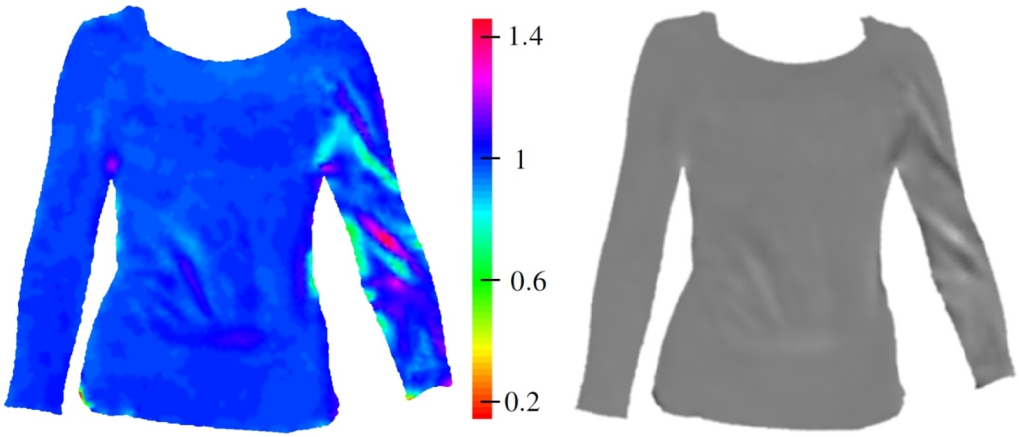


Figure 3.3.: Distribution of photometric parameters on a mesh (left) and generated shading map (right).

With such a photometric warp, the brightness constancy assumption in Equ. (3.24) can be relaxed allowing for multiplicative deviations from brightness constancy:

$$\mathcal{W}_p(\mathbf{x};\boldsymbol{\theta}_p) \cdot \mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x};\boldsymbol{\theta}_s)\big) = \mathcal{I}_\mathcal{T}(\mathbf{x}) \ . \tag{3.48}$$

As the target is to capture complex local shading effects, the photometric warp is modeled as a mesh-based warp akin to the spatial warp. Any other parametrization can be used instead, depending on the given problem, e.g. a global scale or an affine transformation of the color space.

Recall that the mesh-based spatial warp $\mathcal{W}_s$ is parameterized by a mesh $\mathcal{M}_\mathcal{S} : \{\mathbf{V}, \mathcal{F}\}$ on the source image $\mathcal{I}_\mathcal{S}$ and vertex displacements $\Delta\mathbf{V}$ to corresponding vertex positions in the target image $\mathcal{I}_\mathcal{T}$ (Equ. (3.34)-(3.36)):

$$\begin{aligned} &\mathcal{W}_s : \{\mathbf{V}, \mathcal{F}, \Delta\mathbf{V}\} \\ &\Delta\mathbf{V} = [\Delta\mathbf{u}\ \Delta\mathbf{v}] \ . \end{aligned}$$

The warp function is defined in Equ. (3.28), and a parameterization for mesh-based models is given in Equ. (3.39)-(3.40).

A mesh-based photometric warp can now be defined accordingly by the same mesh on the source image and one intensity scale parameter ρ_k per vertex, concatenated to a vector $\boldsymbol{\rho}$:

$$\begin{aligned} &\mathcal{W}_p : \{\mathbf{V}, \mathcal{F}, \boldsymbol{\rho}\} \\ &\boldsymbol{\rho} = [\rho_1\ \ldots\ \rho_K]^T \ . \end{aligned} \tag{3.49}$$

Then, the photometric parameter vector equals the intensity scale parameters of all vertices:

$$\boldsymbol{\theta}_p = \boldsymbol{\rho} \ . \tag{3.50}$$

For a triangle mesh with barycentric parametrization, the warp definition is as follows. Suppose, $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ denote the vertices of the mesh triangle surrounding

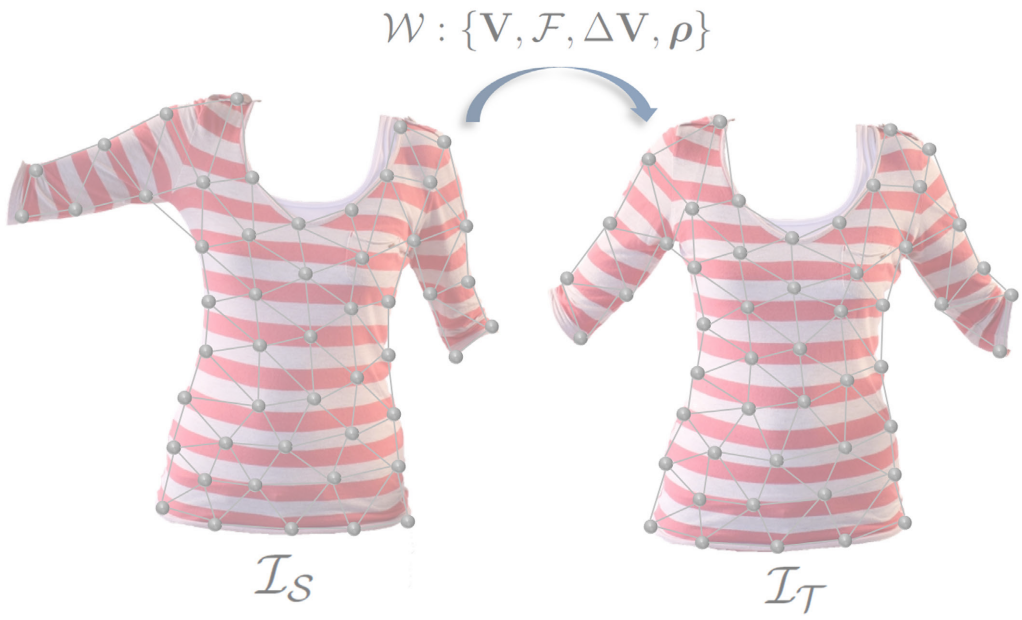


Figure 3.4.: A spatial image warp between $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$ can be defined by a mesh on the source image and vertex displacements to corresponding vertex positions in the target image. A photometric warp to model shading differences (e.g. at the right arm) can be defined by the same mesh and one intensity scale parameter ρ per vertex (see also Fig. 3.3).

pixel $\mathbf{x}_i$ and $\beta_1, \beta_2, \beta_3$ are the corresponding barycentric coordinates. A mesh-based photometric warp can then be defined as

$$\mathcal{W}_p\left(\mathbf{x}_i; \boldsymbol{\theta}_p\right) = \sum_{l=1}^{3} \beta_l \rho_l = \mathbf{B}_p(\mathbf{x}_i) \cdot \boldsymbol{\theta}_p \ , \tag{3.51}$$

where $\mathbf{B}_p(\mathbf{x}_i) = \mathbf{b}(\mathbf{x}_i)$ is a $1 \times K$ warp parametrization matrix with $\mathbf{b}(\mathbf{x}_i)$ as defined in Equ. (3.40). The Jacobians of the spatial and the photometric warps, which are needed in the optimization (Equ. (3.56)), are given by

$$\begin{aligned} \mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i; \boldsymbol{\theta}_s) &= \mathbf{B}_s(\mathbf{x}_i) \\ \mathbf{J}_{\mathcal{W}_p}(\mathbf{x}_i; \boldsymbol{\theta}_p) &= \mathbf{B}_p(\mathbf{x}_i) \ . \end{aligned} \tag{3.52}$$

Finally, a joint spatial and photometric mesh-based warp $\mathcal{W}(\mathcal{I}_\mathcal{S}) = \mathcal{W}_p\big(\mathcal{I}_\mathcal{S}\big(\mathcal{W}_s(\mathbf{x})\big)\big)$ that maps $\mathcal{I}_\mathcal{S}$ onto $\mathcal{I}_\mathcal{T}$ both in the spatial as well as in the intensity domain can be defined by the mesh on the source image, vertex displacements to the corresponding vertex positions in the target images plus a vector of the intensity scale parameters per vertex (Fig. 3.4):

$$\mathcal{W} : \{\mathbf{V}, \mathcal{F}, \Delta\mathbf{V}, \boldsymbol{\rho}\} \ . \tag{3.53}$$

For a mesh with K vertices, a joint warp has $N = N_s + N_p = 3K$ parameters (two spatial and one photometric parameter per vertex) and can be interpreted as a 3D mesh-based warp in the $xy\mathcal{I}$-space with three parameters $\{\Delta u_k, \Delta v_k, \rho_k\}$ per vertex $\mathbf{v}_k$.

The data term $\mathcal{E}_D(\boldsymbol{\theta})$ for the relaxed brightness constancy assumption in Equ. (3.48) is given by the difference between the spatially and photometrically warped source image and the target image, summed up over all image pixels in the region of interest $\mathcal{R}$:

$$\begin{aligned} \mathcal{E}_D(\boldsymbol{\theta}) &= \sum_{\mathbf{x}_i \in \mathcal{R}} \psi\big(r_i(\boldsymbol{\theta})\big) \\ r_i(\boldsymbol{\theta}) &= \mathcal{W}_p(\mathbf{x}_i; \boldsymbol{\theta}_p) \cdot \mathcal{I}_{\mathcal{S}}\big(\mathcal{W}_s(\mathbf{x}_i; \boldsymbol{\theta}_s)\big) - \mathcal{I}_{\mathcal{T}}\left(\mathbf{x}_i\right) \ . \end{aligned} \tag{3.54}$$

In this equation, the $N \times 1$ parameter vector

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_s \\ \boldsymbol{\theta}_p \end{bmatrix} \tag{3.55}$$

comprises both the spatial parameters $\boldsymbol{\theta}_s$ and the photometric parameters $\boldsymbol{\theta}_p$. For the optimization (Gauss-Newton or Levenberg-Marquardt, Sec. 3.1.1), the Jacobian $\mathbf{J_r}$ of the residuals $r_i(\boldsymbol{\theta})$ are needed. The Jacobian $\mathbf{J_r}$ is a $P \times N$ matrix with the following i^{th} row for pixel $\mathbf{x}_i$ (compare Equ. (3.26)):

$$\begin{aligned} \frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} &= \left[\frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_s} \ \frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_p} \right] \\ \frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_s} &= \mathcal{W}_p(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p) \cdot \Big(\nabla \mathcal{I}_{\mathcal{S}}\big(\mathcal{W}_s(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)\big)^T \cdot \mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s) \Big) \\ \frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_p} &= \mathcal{I}_{\mathcal{S}}\big(\mathcal{W}_s(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)\big) \cdot \mathbf{J}_{\mathcal{W}_p}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p) \ . \end{aligned} \tag{3.56}$$

P is again the number of pixels in the region of interest $\mathcal{R}$, and N_s and N_p denote the number of the spatial and photometric warp parameters. $\mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)$ and $\mathbf{J}_{\mathcal{W}_p}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p)$ are the spatial and photometric warp Jacobians at pixel $\mathbf{x}_i$, evaluated at the current parameter estimates $\hat{\boldsymbol{\theta}}_s$, $\hat{\boldsymbol{\theta}}_p$.

In the joint warp model, each vertex holds three parameters $\{\Delta u_k, \Delta v_k, \rho_k\}$, i.e. two spatial parameters and one photometric parameter. The smoothness term is defined as in Equ. (3.41) by

$$\begin{aligned} \mathcal{E}_S(\boldsymbol{\theta}) &= \sum_k \psi\big(s_k(\boldsymbol{\theta})\big) \\ \mathbf{s}(\boldsymbol{\theta}) &= \boldsymbol{\Gamma} \cdot \boldsymbol{\theta} \ , \end{aligned}$$

with a Tikhonov matrix $\boldsymbol{\Gamma}$ penalizing both the spatial deformation field as well as the intensity scale field:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_p \mathbf{L} \end{bmatrix} \ . \tag{3.57}$$

λ_p weights the smoothing of the photometric parameters against the smoothing of the geometric parameters. This is necessary because of the different scaling of the spatial and photometric parameters, the former being an additive parameter, the latter being a multiplicative parameter.

3.2.2. Extension to Color Images

This section proposes an extension of the previously introduced photometric warp to color images. In this section, $\mathcal{I}(\mathbf{x}) = \left[\mathcal{I}^R(\mathbf{x})\ \mathcal{I}^G(\mathbf{x})\ \mathcal{I}^B(\mathbf{x})\right]^T$ denotes a color image, represented by a red, green and blue channel. Let $\mathcal{W}_{pc}\left(\mathcal{I}(\mathbf{x}); \boldsymbol{\theta}_{pc}\right)$ be a photometric warp for color images, locally altering the intensities of each color channel:

$$\begin{aligned}\mathcal{W}_{pc}(\mathcal{I}(\mathbf{x}); \boldsymbol{\theta}_{pc}) &\equiv \mathcal{W}_{pc}(\mathbf{x}; \boldsymbol{\theta}_{pc}) \circ \mathcal{I}(\mathbf{x}) \\ &= \begin{bmatrix} \mathcal{W}_p^R\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \\ \mathcal{W}_p^G\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \\ \mathcal{W}_p^B\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \end{bmatrix} \circ \begin{bmatrix} \mathcal{I}^R(\mathbf{x}) \\ \mathcal{I}^G(\mathbf{x}) \\ \mathcal{I}^B(\mathbf{x}) \end{bmatrix} \ .\end{aligned} \tag{3.58}$$

$\mathcal{W}_p^R\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right)$, $\mathcal{W}_p^G\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right)$ and $\mathcal{W}_p^B\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right)$ denote the photometric warps of the red, green and blue color channels of the image. $\boldsymbol{\theta}_{pc}$ is the photometric parameter vector for color images (defined below in Equ. (3.61)) and $\circ$ denotes the Hadamard product, i.e. the entrywise product of the color values with the respective warp.

For color images, the relaxed brightness constancy assumption in Equ. (3.48) is changed to

$$\mathcal{W}_{pc}(\mathbf{x}; \boldsymbol{\theta}_{pc}) \circ \mathcal{I}_{\mathcal{S}}\big(\mathcal{W}_s(\mathbf{x}; \boldsymbol{\theta}_s)\big) = \mathcal{I}_{\mathcal{T}}\left(\mathbf{x}\right) \ . \tag{3.59}$$

One possible warp parameterization would be to determine a full photometric warp for each color separately. However, this would increase the number of parameters by twice the number of mesh vertices because three intensity scale parameters ρ^R, ρ^G and ρ^B would be necessary for each vertex. Another model is to assume that intensities vary locally, but color changes are spatially global. This can be expressed by

$$\begin{aligned}\mathcal{W}_{pc}\left(\mathbf{x}_i; \boldsymbol{\theta}_{pc}\right) &= \begin{bmatrix} \mathcal{W}_p^R\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \\ \mathcal{W}_p^G\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \\ \mathcal{W}_p^B\left(\mathbf{x}; \boldsymbol{\theta}_{pc}\right) \end{bmatrix} \\ &= \mathcal{W}_p\left(\mathbf{x}_i; \boldsymbol{\theta}_p\right) \cdot \begin{bmatrix} \gamma_{rg} \\ 1 \\ \gamma_{bg} \end{bmatrix} \\ &= \mathcal{W}_p\left(\mathbf{x}_i; \boldsymbol{\theta}_p\right) \cdot \boldsymbol{\gamma}(\boldsymbol{\theta}_\gamma) \ ,\end{aligned} \tag{3.60}$$

with the color gain function $\boldsymbol{\gamma}(\boldsymbol{\theta}_\gamma) = \left[\gamma_{rg}\ 1\ \gamma_{bg}\right]^T$ and $\boldsymbol{\theta}_\gamma = \left[\gamma_{rg}\ \gamma_{bg}\right]^T$. γ_{rg} and γ_{bg} denote global red and blue gains. In this model, $\boldsymbol{\theta}_p$ describes the intensity changes in the green channel that can vary spatially in the image, and $\boldsymbol{\theta}_\gamma$ contains the red and blue gain with respect to the green intensity. These parameters model color changes between two images but are global parameters for one image, i.e. color changes are assumed to be spatially constant. The green channel is used as reference channel because many cameras use a Bayer filter with 50% green but only 25% red and 25% blue pixels.

With this model, the parameter vector $\boldsymbol{\theta}$ is extended by only two additional color gain parameters $\boldsymbol{\theta}_\gamma$ such that for a mesh with K vertices the total number of parameters becomes $N = 3K + 2$:

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_s \\ \boldsymbol{\theta}_p \\ \boldsymbol{\theta}_\gamma \end{bmatrix} \ . \tag{3.61}$$

The data term of the relaxed color constancy assumption in Equ. (3.59) is now given by summing up all pixelwise errors over the three color channels $c \in \{R, G, B\}$ (compare Equ. (3.54)):

$$\begin{aligned} \mathcal{E}_D(\boldsymbol{\theta}) &= \sum_{c \in \{R,G,B\}} \sum_{\mathbf{x}_i \in \mathcal{R}} \psi\big(r_i^c(\boldsymbol{\theta})\big) \\ r_i^c &= \mathcal{W}_p^c(\mathbf{x}_i; \boldsymbol{\theta}_{pc}) \cdot \mathcal{I}_\mathcal{S}^c\big(\mathcal{W}_s(\mathbf{x}_i; \boldsymbol{\theta}_s)\big) - \mathcal{I}_\mathcal{T}^c(\mathbf{x}_i) \\ &= \mathcal{W}_p(\mathbf{x}_i; \boldsymbol{\theta}_p) \cdot \gamma^c(\boldsymbol{\theta}_\gamma) \cdot \mathcal{I}_\mathcal{S}^c\big(\mathcal{W}_s(\mathbf{x}_i; \boldsymbol{\theta}_s)\big) - \mathcal{I}_\mathcal{T}^c(\mathbf{x}_i) \ , \end{aligned} \tag{3.62}$$

where $\gamma^c(\boldsymbol{\theta}_\gamma)$ denotes the entry of the color gain function according to the color channel c.

Each pixel now contributes three equations to the Jacobian $\mathbf{J_r}$, one for each color channel (compare Equ. (3.26) and (3.56)):

$$\begin{aligned} \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}} &= \left[\frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_s} \ \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_p} \ \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_\gamma} \right] \\ \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_s} &= \mathcal{W}_p(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p) \cdot \gamma^c(\hat{\boldsymbol{\theta}}_\gamma) \cdot \Big(\nabla \mathcal{I}^c\big(\mathcal{W}_s(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)\big)^T \cdot \mathbf{J}_{\mathcal{W}_s}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s) \Big) \\ \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_p} &= \gamma^c(\hat{\boldsymbol{\theta}}_\gamma) \cdot \mathcal{I}^c\big(\mathcal{W}_s(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)\big) \cdot \mathbf{J}_{\mathcal{W}_p}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p) \\ \frac{\partial r_i^c(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}_\gamma} &= \mathcal{W}_p(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_p) \cdot \mathcal{I}^c\big(\mathcal{W}_s(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_s)\big) \cdot \mathbf{J}_{\boldsymbol{\gamma}}^c(\hat{\boldsymbol{\theta}}_\gamma) \ . \end{aligned} \tag{3.63}$$

$\mathbf{J}_{\boldsymbol{\gamma}}^c(\hat{\boldsymbol{\theta}}_\gamma)$ denotes the row in the color gain Jacobian according to the color channel c. The color gain Jacobian is given by

$$\mathbf{J}_{\boldsymbol{\gamma}}(\boldsymbol{\theta}_\gamma) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \ . \tag{3.64}$$

The smoothness term for the joint spatial and color model is defined as in Equ. (3.41) with a Tikhonov matrix $\boldsymbol{\Gamma}$ as given in Equ. (3.65). The color gain parameters $\boldsymbol{\theta}_\gamma$ are global parameters for the complete mesh and are regularized by penalizing extreme changes:

$$\boldsymbol{\Gamma} = \left[\begin{array}{ccc|c} \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_p \mathbf{L} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_\gamma \mathbf{I}_2 \end{array}\right] \ . \tag{3.65}$$

λ_γ weights the regularization of the global gain parameters against the other regularization terms.

3.2.3. Putting Everything Together: The Optimization Framework

Having defined the data term $\mathcal{E}_D(\boldsymbol{\theta})$ (Equ. (3.54) and (3.62)) and the smoothness term $\mathcal{E}_S(\boldsymbol{\theta})$ (Equ. (3.41)), Quasi-Newton optimization methods can now be used to minimize the error function

$$\mathcal{E}(\boldsymbol{\theta}) = \mathcal{E}_D(\boldsymbol{\theta}) + \lambda^2 \mathcal{E}_S(\boldsymbol{\theta}) \ .$$

For a least-squares (LS) estimator $\psi_{\text{LS}}(\mathbf{r}) = \frac{1}{2} r_i^2$, the gradient and Hessian of $\mathcal{E}(\boldsymbol{\theta})$ are given by

$$\begin{aligned} \mathbf{g}_\mathcal{E}(\boldsymbol{\theta}) &= \mathbf{J}_\mathbf{r}^T(\boldsymbol{\theta})\mathbf{r}(\boldsymbol{\theta}) + \lambda^2 \mathbf{J}_\mathbf{s}^T(\boldsymbol{\theta})\mathbf{s}(\boldsymbol{\theta}) = \mathbf{J}^T(\boldsymbol{\theta})\mathbf{b}(\boldsymbol{\theta}) \\ \mathbf{H}_\mathcal{E}(\boldsymbol{\theta}) &\approx \mathbf{J}_\mathbf{r}^T(\boldsymbol{\theta})\mathbf{J}_\mathbf{r}(\boldsymbol{\theta}) + \lambda^2 \mathbf{J}_\mathbf{s}^T(\boldsymbol{\theta})\mathbf{J}_\mathbf{s}(\boldsymbol{\theta}) = \mathbf{J}^T(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta}) \ , \end{aligned} \tag{3.66}$$

with

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{J}_\mathbf{r}(\hat{\boldsymbol{\theta}}) \\ \lambda \mathbf{J}_\mathbf{s}(\hat{\boldsymbol{\theta}}) \end{bmatrix}, \ \mathbf{b}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{r}(\hat{\boldsymbol{\theta}}) \\ \lambda \mathbf{s}(\hat{\boldsymbol{\theta}}) \end{bmatrix} \ . \tag{3.67}$$

$\mathbf{r} = \mathbf{r}(\hat{\boldsymbol{\theta}})$ is the residual vector of the data term, evaluated at the current parameter estimation $\hat{\boldsymbol{\theta}}$, and $\mathbf{s} = \mathbf{s}(\hat{\boldsymbol{\theta}})$ is the residual vector of the smoothness constraints. The Gauss-Newton (GN) parameter update is given by solving the normal equations

$$\delta\boldsymbol{\theta}_{\text{GN}} = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{b} \ , \tag{3.68}$$

with $\mathbf{J} = \mathbf{J}(\hat{\boldsymbol{\theta}})$ and $\mathbf{b} = \mathbf{b}(\hat{\boldsymbol{\theta}})$. Similarly, the Levenberg (L) and Levenberg-Marquardt (LM) methods find the parameter update by

$$\begin{aligned} \delta\boldsymbol{\theta}_{\text{L}} &= -(\mathbf{J}^T\mathbf{J} + \alpha\mathbf{I})^{-1}\mathbf{J}^T\mathbf{b} \\ \delta\boldsymbol{\theta}_{\text{LM}} &= -\big(\mathbf{J}^T\mathbf{J} + \alpha \cdot \text{diag}(\mathbf{J}^T\mathbf{J})\big)^{-1}\mathbf{J}^T\mathbf{b} \ . \end{aligned} \tag{3.69}$$

In each iteration, the parameter vector is updated by $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}$. For the mesh-based warps, the structure of the Hessian is sparse, which can be exploited to solve the normal equations (Fig. 3.5).

To make the parameter estimation more robust against outliers, a robust estimator (the Huber norm) is exploited instead of the least-squares estimator. This generalized optimization problem can be solved by an iterative reweighted least-squares method (Sec. 3.1.1). In this iterated reweighted scheme, the gradient and the Hessian of the cost function are given by

$$\begin{aligned} \mathbf{g}_\mathcal{E}(\boldsymbol{\theta}) &= \mathbf{J}_\mathbf{r}^T\mathbf{W}_\mathbf{r}\mathbf{r} + \lambda^2 \mathbf{J}_\mathbf{s}^T\mathbf{W}_\mathbf{s}\mathbf{s} \\ \mathbf{H}_\mathcal{E}(\boldsymbol{\theta}) &\approx \mathbf{J}_\mathbf{r}^T\mathbf{W}_\mathbf{r}\mathbf{J}_\mathbf{r} + \lambda^2 \mathbf{J}_\mathbf{s}^T\mathbf{W}_\mathbf{s}\mathbf{J}_\mathbf{s} \ , \end{aligned}$$

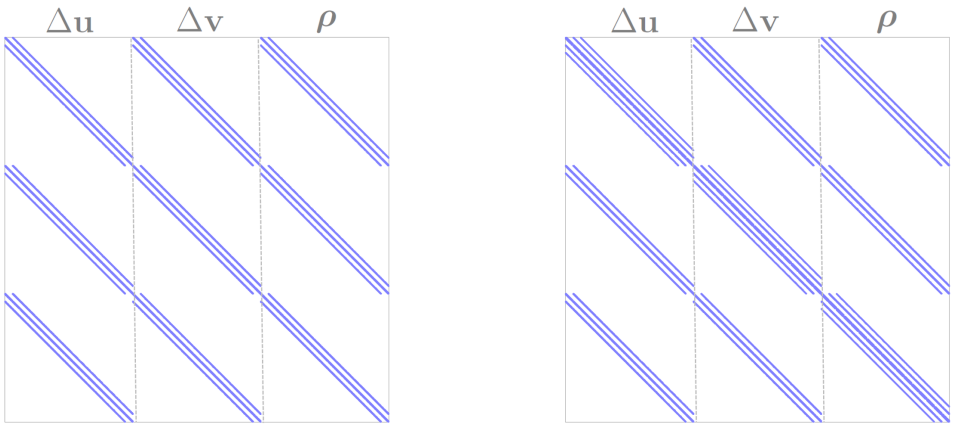


Figure 3.5.: Sparse structure of the Hessian of the data term $\mathbf{H}_{\mathcal{E}_D}(\boldsymbol{\theta}) \approx \mathbf{J}_{\mathbf{r}}^T(\boldsymbol{\theta})\mathbf{J}_{\mathbf{r}}(\boldsymbol{\theta})$ (left) and the combined error function $\mathbf{H}_{\mathcal{E}}(\boldsymbol{\theta}) \approx \mathbf{J}^T(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta})$ (right), consisting of the data and the smoothness term, for a mesh-based joint spatial and photometric model.

where $\mathbf{W_r} = \mathrm{diag}\big(w\big(\mathbf{r}(\hat{\boldsymbol{\theta}})\big)\big)$ is a diagonal weight matrix whose elements are computed from the weight function $w(\mathbf{r}\big(\boldsymbol{\theta}\big)\big)$ as given in Equ. (3.17) in each iteration, and $\mathbf{W_s}$ is defined accordingly. The Gauss-Newton parameter update is then given by

$$\delta\boldsymbol{\theta}_{\mathrm{rGN}} = -(\mathbf{J}^T\ \mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{b}\ , \tag{3.70}$$

with

$$\mathbf{W} = \begin{bmatrix} \mathbf{W_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{W_s} \end{bmatrix}\ .$$

The updates for the robust Levenberg or Levenberg-Marquardt updates are accordingly.

3.2.4. Implementation Details

Initialization. Generally, if not stated differently, the spatial parameter vector is initialized with

$$\boldsymbol{\theta}_{s0} = \mathbf{0}\ ,$$

and the photometric parameters are initialized with unit scales

$$\begin{aligned} \boldsymbol{\theta}_{p0} &= \mathbf{1} \\ \boldsymbol{\theta}_{\gamma 0} &= \mathbf{1}\ . \end{aligned}$$

In some cases, if the motion between the images is large, the spatial warp is initialized with displacements calculated from sparse correspondences, e.g. feature matches, estimated in a previous step. The source of these correspondences will be explained in the context. Let $\mathbf{b}_{\mathcal{S}i}$ and $\mathbf{b}_{\mathcal{T}i}$ denote such sparse corresponding points in the source and in the target images. The source points are not necessarily located at vertex

positions but have a unique position in the mesh $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ on the source image, defined by the barycentric coordinates with respect to the surrounding triangle:

$$\mathbf{b}_{\mathcal{S}i} = \sum_{l=1}^{3} \beta_l \mathbf{v}_l \ . \tag{3.71}$$

Here, $\mathbf{v}_l$ denotes the three vertices of the surrounding triangle, and β_l are the corresponding barycentric coordinates. Vertex displacements that best describe the sparse displacements $\mathbf{d}_i = \mathbf{b}_{\mathcal{T}i} - \mathbf{b}_{\mathcal{S}i}$ should fulfill:

$$\mathbf{d}_i = \sum_{l=1}^{3} \beta_l \Delta\mathbf{v}_l \ . \tag{3.72}$$

As the sparse correspondences can also contain some errors, the mesh Laplacian is used as an additional smoothness constraint, yielding the following linear system (compare App. A.1.3):

$$\begin{bmatrix} \mathbf{D} \\ \lambda\mathbf{L} \end{bmatrix} \Delta\mathbf{V} = \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \ , \tag{3.73}$$

with $\Delta\mathbf{V}$ as given in Equ. (3.35). $\mathbf{d}_x$ and $\mathbf{d}_y$ are the column vectors of all sparse displacements in $x-$ and $y-$direction and $\mathbf{D}$ is a data matrix with one $1 \times K$ row vector $\mathbf{p}(\mathbf{b}_{\mathcal{S}i})$ per point $\mathbf{b}_{\mathcal{S}i}$, with entries:

$$p_j(\mathbf{b}_{\mathcal{S}i}) = \begin{cases} \beta_l & \text{if } \mathbf{v}_j \text{ is the } l^{th} \text{ vertex in the triangle surrounding } \mathbf{b}_{\mathcal{S}i} \\ 0 & \text{otherwise} \end{cases} \ . \tag{3.74}$$

This interpolation scheme yields smooth yet detail preserving interpolation of the feature disparities. The interpolated vertex displacements $\Delta\mathbf{V}$ provide an initialization to the warp optimization with $\boldsymbol{\theta}_{s0} = \left[\Delta\mathbf{u}^T \ \Delta\mathbf{v}^T\right]^T$.

Robust Estimation. Most results and experiments presented in this dissertation exploit the Huber estimator given in Equ. (3.19) during optimization. The Huber function does not reject outliers, i.e. residuals with $|r_i| > \kappa_{\text{H}}$, completely but rather gives outliers less weight than the least-squares estimator. The threshold κ_{H} has to be chosen carefully. If κ_{H} is chosen too large, outliers might not be detected; if κ_{H} is chosen too small, a lot of information is lost because too many data points are detected as outliers. One approach to determine a value for κ_{H} depending on the data, is to use the median absolute deviation (MAD) of the residuals, which is an efficient score for outlier rejection [HMT00] (App. A.1.2).

Let $\tilde{r}$ denote the median of the residuals. The MAD is defined as

$$\text{MAD}(\mathbf{r}) = \text{median} \left|\mathbf{r} - \tilde{r}\right| \ ,$$

and the MAD score for a residual r_i is calculated by

$$M_i = \tfrac{r_i - \tilde{r}}{\text{MAD}(\mathbf{r})} \ .$$

Residuals r_i are rejected if $|M_i| > t$, where t is the maximum permissible MAD score. For a large number of data points and Normal distribution, a value of $t = 3.5$ is suggested in the literature [IH93].

The Huber function (Equ. (3.19)) is a parabola in the vicinity of zero and increases linearly at a given level $|r_i| > \kappa_{\text{H}}$. Therefore, it will be assumed that the median of the residuals is $\tilde{r} = 0$ because otherwise the kernel would be shifted by the median. A MAD-based determination of the threshold for the Huber function is then given by:

$$\begin{aligned} \left| \frac{r_i}{\text{MAD}(\mathbf{r})} \right| > t \Rightarrow |r_i| &> t \cdot |\text{MAD}(\mathbf{r})| \\ \Rightarrow \kappa_{\text{H}} &= t \cdot |\text{MAD}(\mathbf{r})| \ . \end{aligned} \tag{3.75}$$

Hierarchical Framework. In the presented framework, the optimization is performed hierarchically on an image pyramid where each level yields a more and more accurate parameter estimate. On each level, the regularization parameter λ is adapted, starting with larger values on the smallest pyramid level, favoring smooth warps, and decreasing values for the following levels, resulting in a coarse-to-fine framework. This hierarchical framework has several advantages. It speeds up the iteration time on lower and coarser levels and allows coping with large displacements. If computation time is not important, it is also a good strategy to iterate over several regularization parameters on each level. In practice, $3 - 4$ pyramid levels have been used in the experiments.

3.3. Applications and Experimental Evaluation

3.3.1. Applications

Several applications –and results in that context– of the presented joint spatial and photometric warp optimization approach are presented in the following and the remainder of this thesis:

Image-Based Retexturing. An application of the proposed warp optimization approach to image-based retexturing is presented in Chapter 5. Image-based retexturing means that a texture in an image is exchanged by a synthetic one while texture deformation and shading properties are preserved (Fig. 5.1 on page 94). If a reference of the undeformed and uniformly lit texture is available, the extraction of texture deformation and shading can be formulated as a joint spatial and photometric warp estimation task between the original and the reference image. The new synthetic texture is then spatially and photometrically warped with the estimated warp parameters and blended into the original image. For this application, the extraction of local photometric parameters is essential to establish detailed shading maps for the synthesis of a realistic and visually correct augmented result (see e.g. Fig. 5.9 on page 105 and Fig. 5.10-5.12 on pages 106-107).

Figure 3.6.: Augmenting deformable surfaces in single-view video sequences under varying lighting conditions: original frame (left) and augmented example frames.

Non-Rigid Tracking and Video Augmentation. The presented warp optimization method can be exploited to robustly track (and augment) deforming surfaces in monocular videos under changing lighting conditions. *Tracking* in this case, refers to the estimation of non-rigid deformations and motion. Because intensity-based tracking is known to suffer from drift and error accumulation when performed in a frame-to-frame approach, it is often performed in an analysis-by-synthesis approach (page 49). In this approach, the same image (e.g. the first video frame) serves as a reference for all video frames to avoid misalignment of the mesh-based model and the reference image. However, this approach is sensitive to brightness changes because differences between the reference image and the current frame can increase over the sequence. If intensity changes are not considered in the warp model, the spatial tracking can get lost. The estimation of photometric warps clearly improves the spatial tracking results, making the warp optimization more robust against lighting changes (see Fig. 3.10 on page 51). The following section evaluates the approach for tracking purposes quantitatively.

Furthermore, the extracted warp parameters can be used to augment deformable surfaces in single-view video sequences [HE09c, HE09a, HSE10] (Fig. 3.6). Assuming that the first frame of a sequence shows an image of the undeformed and uniformly lit surface, the extracted warp parameters yield information about absolute texture deformation and shading and can directly be applied to an arbitrary virtual texture, which is realistically augmented onto the moving surface. When tracking and augmenting a deforming surface in monocular video, special attention has to be payed to external occlusions as well as self-occlusions, which appear if the surface itself is deformed in such a way that parts of it occlude other parts (Fig. 3.6, top row).

Figure 3.7.: Realization of a Virtual Mirror prototype. A user can select between various designs on a touchscreen and can then see himself with the virtually textured clothing while moving freely in front of the system.

Figure 3.8.: Retexturing in the Virtual Mirror: original camera image with mesh-based warp model (left) and augmented results shown in the mirror.

Occlusion handling is out of the scope of this thesis, but the interested reader is referred to [HE08, HSE10], where strategies for external and self-occlusion handling are described in detail. Fig. 3.6 shows augmented frames of three example video sequences. These examples demonstrate how the virtual texture follows the movements and deformations of the original surface and complex lighting and shading changes from the original video are maintained in the augmented result.

The video augmentation approach has been implemented into a real-time prototype of a Virtual Mirror (Fig. 3.7-3.8) [HE09c]. In this prototype system, the mirror is replaced by a large display, which shows the mirrored image of a camera capturing the upper part of the body of a person. The system changes the logo and the color of a user's shirt in the displayed image while the user can move freely in front of the mirror wearing a prototype shirt. On a touch screen, the user can select between different colors and logos. To provide an undeformed view of the region of interest on his shirt, the user is asked to stand in a frontal pose to the camera before the tracking and augmentation starts. By tracking the elastic deformations and recovering a shading map in real time, the chosen texture is realistically augmented onto the moving shirt such that the user's reflection in the mirror seems to be wearing the virtually textured shirt. Fig. 1.1 on page 5 shows the concept and design of the system. Fig. 3.7 shows the realized system, and Fig. 3.8 depicts different retexturing

examples of a sequence captured with the Virtual Mirror system.

The Virtual Mirror system was demonstrated at several exhibitions (e.g. IFA 2008, CeBIT 2009, Kiosk 2009, Siggraph 2009, CeTech 2009-2012, RETAILTECH JAPAN 2013), where it was tested by several users unfamiliar with the system. The system was honored as Selected Landmark 2009 (Ort im Land der Ideen 2009) by the Germany-Land of Ideas initiative. In the same year the European Association for Self- Service singled out the Virtual Mirror as the most innovative product displayed at the international trade fair Kiosk Europe 2009.

Image-Based Rendering in Pose-Space. Chapter 4 introduces a new pose-dependent image-based rendering approach. To interpolate new images from a previously recorded multi-view/multi-pose database of images (Fig. 4.16-4.17 on pages 79-80), pose-dependent appearance is extracted from the images as spatial and photometric warps between them, estimated with the warp estimation approach presented in this chapter. For rendering, the warp parameters are interpolated in the space of body poses. When body pose is changed, clothes not only change their shape, but also the shading pattern varies because of new wrinkles and creases. Shading differences in the images are accounted for by interpolating not only spatial but also photometric warps (see e.g. Fig. 4.7 on page 66).

Depth Estimation. For the establishment of pose-dependent image-based databases of clothing appearance in Chapter 4, the presented warp optimization method is exploited to estimate the coarse 3D shape of pieces of clothing from stereo correspondences in a multi-view setup (Fig. 4.4 on page 63). For this application, the spatial warp is initialized with correspondences from a sparse reconstruction achieved with the method of [SSS08] (Sec. 4.1.2). Although the presented mesh-based approach is not specifically designed for 3D reconstruction and rather estimates piecewise affine warps, it achieves good refinement results of the sparse reconstruction. In fact, if the mesh triangles are small enough, they can be modeled as planar patches, and an affine transformation is a good approximation for a homography between the planar patches [HSE11b]. The reconstructed geometries are used to guide the warping of database images for pose-dependent image-based rendering.

3.3.2. Experimental Evaluation

For quantitative evaluation, the proposed joint spatial and photometric warp optimization approach was applied to various non-rigid image registration and tracking tasks, presented in the following. Tab. 3.1 lists the notation for the different warp models and optimization methods used in the following evaluation. In all experiments, all other parameters (e.g. mesh resolution, regularization parameters, resolution of the image pyramid etc.) of results that are directly compared to each other are kept constant to allow an equitable comparison of the results.

Stereo Dataset. The introduced warp optimization method has been applied to register image pairs from the Cloth1-3 stereo images (views 1 and 5) of the Mid-

Category	Abbr.	Meaning
Warp model	s	Spatial warp
	sp	Joint spatial and photometric warp
	spc	Joint spatial and photometric color warp
Optimization kernel	LS	Least-squares
	H	Huber

Table 3.1.: Parameter abbreviations used in the evaluation.

	Disparity			Intensity		
	s	sp	spc	s	sp	spc
Cloth1	0.7190	0.7079	0.7321	0.0135	0.0087	0.0088
Cloth2	1.8984	1.6772	1.7556	0.0187	0.0110	0.0113
Cloth3	1.8825	1.6243	1.6604	0.0185	0.0134	0.0136
Cloth1 Illum1	0.8869	0.8682	0.8718	0.1464	0.0420	0.0270
Cloth1 Illum2	0.7973	0.7390	0.7350	0.1091	0.0461	0.0203
Cloth2 Illum1	11.1836	2.1846	2.2604	0.2463	0.0404	0.0192
Cloth2 Illum2	4.0531	1.4944	1.7008	0.0897	0.0272	0.0164
Cloth3 Illum1	7.9030	2.3332	2.2501	0.1340	0.0439	0.0356
Cloth3 Illum2	2.2205	1.8324	1.9113	0.0463	0.0221	0.0214
mean all	3.5049	**1.4957**	1.5419	0.0914	0.0283	**0.0193**
mean Cloth 1-3	1.5000	**1.3365**	1.3827	0.0169	**0.0110**	0.0112

Table 3.2.: Mean disparity and intensity errors for stereo pairs with provided ground truth disparities for the three different warping models. The datasets were taken from the Middlebury 2006 dataset [HS07] and are described in App. A.2. (H, LM).

dlebury 2006 stereo dataset[2] [HS07] (Tab. 3.2). This dataset provides ground truth disparity maps between the two views. Besides constant illumination situations, the dataset features all scenes and viewpoints captured under three different illuminations (1-3) with three different exposures (0-2). Besides stereo pairs taken under the same illumination, the following experiment also includes registration of stereo pairs captured under different illuminations and exposures (denoted by Illum1 and Illum2 in this thesis). A summary of the used illuminations and exposures is given in App. A.2 (see Tab. A.1 and Fig. A.2 on page 117). Different illuminations in the source and target images cause local intensity differences (different shading patterns because of different light directions), whereas both different illumination and exposure cause a global change in intensity and color (Fig. 3.9), making intensity-based registration challenging because the brightness constancy assumption is not valid. To improve the registration results for the spatial warp model (s), smooth intensity changes in the image were removed by high pass filtering prior to warp optimization. Sparse SIFT [Low03] correspondences were used for warp initialization in all examples (Sec. 3.2.4).

For quantitative evaluation and comparison of the three presented warp models, two

[2] http://vision.middlebury.edu/stereo/data/scenes2006/, downloaded on February 19, 2013

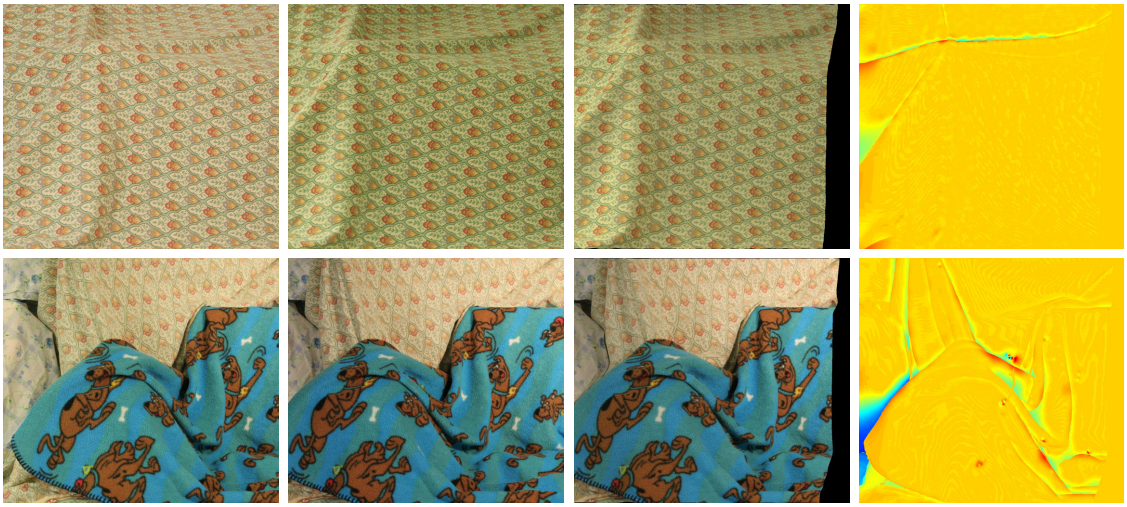


Figure 3.9.: Registration results for Cloth1 Illum2 and Cloth3 Illum2. From left to right: source image (view 1); target image (view 5); warped source image; distribution of the disparity error (from blue colors for negative errors via orange to red for positive errors). The error is depicted both in unoccluded and occluded regions. Note, how local shading and color are adapted in the warped source image. (H, LM, spc).

error measures are listed in Tab. 3.2. The first error measure reports the mean error between the estimated and the ground truth disparities, calculated as

$$E_{\text{Disp}} = \frac{1}{|\mathcal{R}|} \sum_{x \in \mathcal{R}} \left| \hat{\mathcal{D}}(\mathbf{x}) - \mathcal{D}(\mathbf{x}) \right| , \tag{3.76}$$

where $\hat{\mathcal{D}}(\mathbf{x})$ denotes the estimated disparity at pixel position $\mathbf{x}$ and $\mathcal{D}(\mathbf{x})$ denotes the ground truth disparity. The second error measure reports the mean intensity error over all three color channels between the warped source image (view 1) and the target image (view 5):

$$E_{\text{Int}} = \frac{1}{3\,|\mathcal{R}|} \sum_{c \in \{R,G,B\}} \sum_{x \in \mathcal{R}} \left| \hat{\mathcal{I}}^c_{\mathcal{S}}(\mathbf{x}) - \mathcal{I}^c_{\mathcal{T}}(\mathbf{x}) \right| , \tag{3.77}$$

where $\hat{\mathcal{I}}^c_{\mathcal{S}}(\mathbf{x})$ denotes the color channel $c \in \{R, G, B\}$ of the warped source image. The errors are evaluated in non-occluded regions (occlusion maps are generated by cross checking the ground truth disparity maps) and a border region of 10 pixels has been excluded from evaluation.

Tab. 3.2 shows that over all stereo pairs, the mean disparity error is reduced by over 56% by the joint warp models (sp, spc), compared to the spatial warp model (s). The intensity error can be reduced by 69% by the sp warp model and by 78% by the spc warp model. Based on the intensity-based error measure, the spc model outperforms the sp warp model because the Illum1 and Illum2 stereo pairs exhibit different colors due to different illumination and exposures during capturing, not modeled in the sp warp. Even if the mean error is only calulated over the stereo pairs taken under the same illumination conditions (Cloth1-3, bottom line in Tab. 3.2), the

joint warp models can improve the registration results based on both error measures (approximately 33% for the intensity-based error and 8% for the disparity-based error) because different viewpoints cause slightly different shading patterns in the stereo images, violating the brightness constancy assumption in these regions. Here, the optimization of the spc warp model does not improve the registration, compared to the sp warp model, because the global colors in the stereo pair are similar, in contrast to local shading. Fig. 3.9 shows two registration results achieved with the spc warp model. In the warped source image (third image in each row), both global color as well as local shading have been adapted to the target image (second image in each row), according to the estimated warp parameters. The rightmost image in each row depicts the distribution of the intensity-based error in the image. The error is large at depth discontinuities in the scene, mainly because of the mesh-based warp model and the fact that vertices are not necessarily placed at depth discontinuities in the image. Also, the smoothness term prevents too strong discontinuities in the deformation as well as the shading field.

Deformable Registration and Tracking. The presented warp estimation approach has been exploited to follow non-rigid deformations of surfaces and objects in monocular video sequences. The video sequences used for the following analysis are described in App. A.2 (Tab. A.2 on page 118, Fig. A.3 on page 119). While the surfaces deform, lighting and shading on the surface changes because the surface normals change their direction to the light source.

Deformation tracking is performed by consecutively registering a reference frame to all video frames. In all experiments, the first video frame served as a reference image. This reference image is warped to the previous video frame and serves as source image for warp optimization to the current frame. In this analysis-by-synthesis approach, the model frame serves as a reference image for all subsequent frames, thereby avoiding error accumulation and allowing for recovery from small registration errors during tracking.

Let $\mathcal{I}_j, j = 0 \dots N$ denote the frames of a video sequence. Furthermore, let $\mathcal{W}_{(j-1)\to j}$ denote a joint spatial and intensity warp from $\mathcal{I}_{j-1}$ to $\mathcal{I}_j$. $\mathcal{I}_j$ be the frame to be processed and $\mathcal{I}_0$ be the first frame used as model image. The previously estimated parameter sets $\{\Delta\mathbf{V}_1, \dots \Delta\mathbf{V}_{j-1}, \boldsymbol{\rho}_1 \dots \boldsymbol{\rho}_{j-1}\}$ are accumulated to generate a warp

$$\mathcal{W}_{0\to j-1} = \mathcal{W}_{0\to 1} \oplus \mathcal{W}_{1\to 2} \dots \oplus \mathcal{W}_{j-2\to j-1}$$

from the model image $\mathcal{I}_0$ to the previous frame $\mathcal{I}_{j-1}$, where $\oplus$ denotes the concatenation of warp parameters. The new warp parameters $\{\Delta\mathbf{V}_j, \boldsymbol{\rho}_j\}$ are then estimated from a synthetic previous frame $\hat{\mathcal{I}}_{j-1} = \mathcal{W}_{0\to j-1}(\mathcal{I}_0)$, generated by warping the reference frame onto the frame $\mathcal{I}_{j-1}$, to the current frame $\mathcal{I}_j$.

For evaluation of the deformable tracking results, the root mean squared error (RMSE) of the pixelwise intensity difference is calculated between the synthetic frame $\hat{\mathcal{I}}_j = \mathcal{W}_{0\to j}(\mathcal{I}_0)$, generated by warping the reference frame onto frame $\mathcal{I}_j$, and

	s		sp		spc	
	H	LS	H	LS	H	LS
Bedsheet	0.1698	0.1675	0.0655	0.0590	0.0592	0.0481
Cushion	0.1092	0.1331	0.0452	0.0449	0.0323	0.0312
Flowers	0.1525	0.2873	0.0432	0.0406	0.0624	0.0617
Shirt	0.0761	0.0754	0.0646	0.0602	0.0410	0.0390
Paper1	0.0567	0.0664	0.0444	0.0454	0.0283	0.0284
Paper2	0.0920	0.1520	0.0413	0.0436	0.0232	0.0249
Paper3	0.0943	0.0935	0.0385	0.0325	0.0353	0.0409
Testpattern	0.0595	0.0593	0.0203	0.0180	0.0134	0.0131
mean	0.1012	0.1293	0.0454	0.0430	0.0369	0.0359

Table 3.3.: Comparison of the average intensity RMSE over eight video sequences for the three warping models for robust (H) and least-squares (LS) optimization. The video sequences are described in App. A.2.

the actual frame $\mathcal{I}_j$. The RMSE is computed over all image pixels in the mesh region $\mathcal{R}$:

$$E_{\mathrm{RMSE}_j} = \sqrt{\frac{1}{3\,|\mathcal{R}|} \sum_{c\in\{R,G,B\}} \sum_{x\in\mathcal{R}} \left(\hat{\mathcal{I}}_j^c(\mathbf{x}) - \mathcal{I}_j^c(\mathbf{x})\right)^2} \,. \tag{3.78}$$

Tab. 3.3 compares the mean intensity RMSE over all frames of eight video sequences achieved by the three proposed warp models with robust (H) as well as least-squares (LS) optimization:

$$E_{\mathrm{RMSE}} = \frac{1}{N} \sum_{j=1}^{N} E_{\mathrm{RMSE}_j} \,. \tag{3.79}$$

The results show that taking illumination parameters into account significantly reduces the mean intensity RMSE over the entire sequence by more than 55% for robust optimization and more than 66% for least-squares optimization. Fig. 3.10 plots the progress of E_{RMSE_j} over the video frames for four example video sequences and compares results for the three warping models (s red, sp green, spc blue), as well as robust (solid lines) and least-squares optimization (dashed lines). For long video sequences, the analysis-by-synthesis tracking approach is sensitiv to illumination changes in the video if the intensity changes are not accounted for in the warp model. In case of the Paper2 and the Cushion sequences, the spatial model looses track because of increasing intensity differences between the reference frame and the current frame to be analysed, violating the brightness constancy assumption. Here, robust optimization can improve results compared to least-squares optimization. In contrast, for the joint warp models, robust optimization and least-squares optimization yield similar results. The reason for this is the adapted cost function in the data term, which has been modified to better reflect true conditions of changing and complex lighting conditions. This leads to fewer outliers in the data term.
The peaks in the RMSE plot of the Testpattern sequence for the spatial warp model do not result from spatial registration error but rather from the fact that intensities

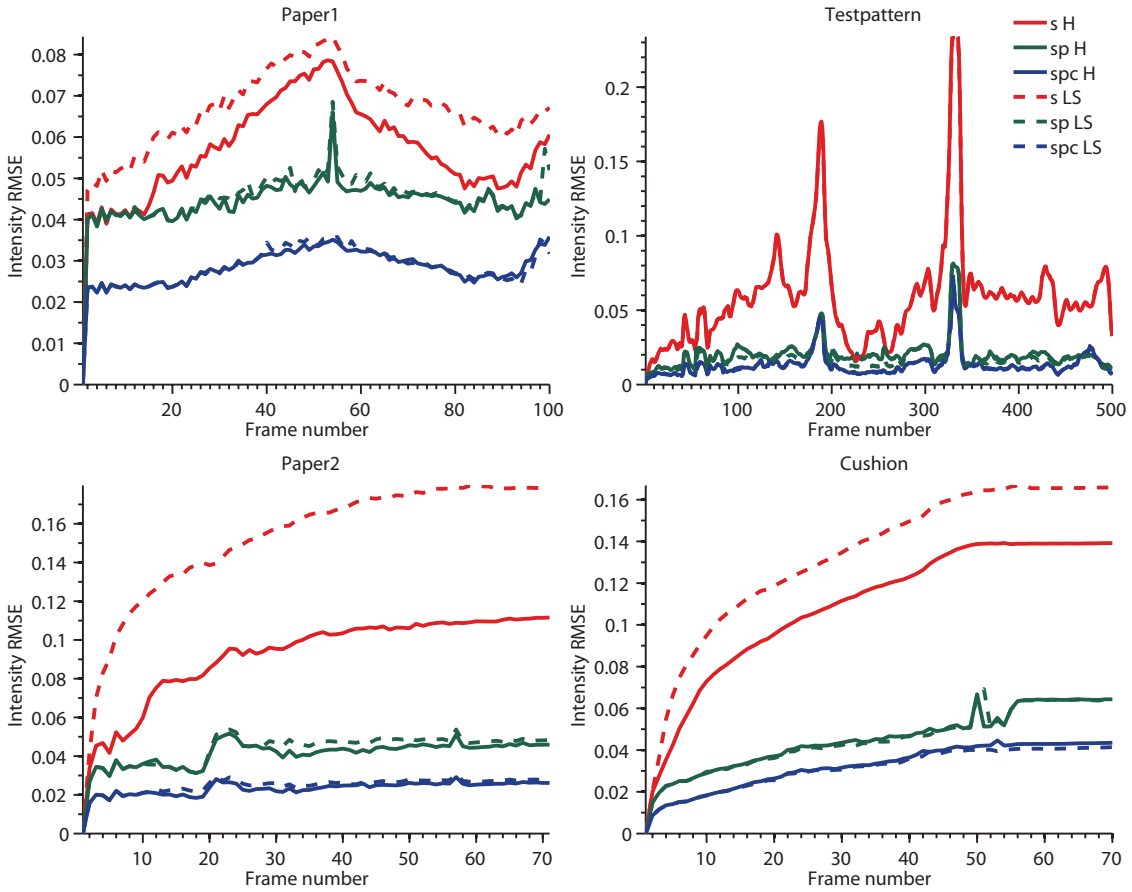
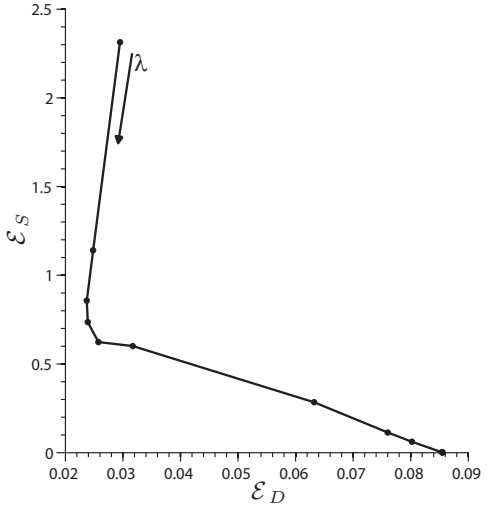


Figure 3.10.: Plots of the intensity RMSE between the synthetic frame $\hat{\mathcal{I}}_j$ and the original current frame $\mathcal{I}_j$ for the three different warping models (s, sp, spc, see Tab. 3.1 on page 47). For comparison, the results achieved with LS optimization (dashed) and robust optimization (solid) are depicted. Plots for all eight video sequences listed in Tab. 3.3 are shown in App. A.2 in Fig. A.4 on page 120. Example frames are shown in Fig. A.3 on page 119.

are not adapted in the warped reference frame. Results for all eight video sequences listed in Tab. 3.3 are shown in App. A.2 in Fig. A.4 on page 120.

Influence of the Regularization Parameter. Choosing the regularization parameter λ in the optimization framework is not a trivial task. The choice of λ is a trade-off between fitting to the data term, which may be corrupted by noise, and the *smoothness* of the result. Hence, there is no *correct* λ, because this trade-off depends on the considered problem. However, a *reasonable* λ represents a good balance between both sides. A convenient graphical tool to analyze the influence of the regularization parameter λ is the L-curve [HO92, ABT05], which plots $\mathcal{E}_S(\hat{\boldsymbol{\theta}})$ versus $\mathcal{E}_D(\hat{\boldsymbol{\theta}})$ for different values of λ. This way, the L-curve clearly displays the compromise between minimization of the data and the regularization term. The L-curve often takes on a characteristic L-shape with a distinct corner, separating the vertical and the horizontal parts of the curve, which correspond to *under-* and *oversmoothed*



λ	$\mathcal{E}_D$	$\mathcal{E}_S$
1000	0.0856	0.0007
500	0.0854	0.0026
100	0.0802	0.0618
75	0.0760	0.1138
50	0.0632	0.2843
25	0.0317	0.6009
15	0.0258	0.6235
10	0.0239	0.7360
7.5	0.0237	0.8577
5	0.0248	1.1406
2.5	0.0295	2.3137

Figure 3.11.: A typical L-curve and associated values for $\mathcal{E}_D$ and $\mathcal{E}_S$ for different values for λ for an image registration task.

solutions (Fig. 3.11). The horizontal part corresponds to solutions where the regularization parameter is large and the cost function is dominated by regularization errors. The vertical part corresponds to solutions where the regularization parameter is small and the cost function is dominated by the data error. The sharpness of the corner varies from problem to problem, but it is usually well-defined. Often, a *reasonable* λ, i.e. a good trade-off between fitting to the data and smoothness term, is considered to be the point of maximal curvature of the L-shaped curve or the point nearest to the origin. Fig. 3.11 shows a typical L-curve and associated values for the data and smoothness terms for different values for λ for an image registration task. For very large values of λ, $\mathcal{E}_D$ is very high as the model is constrained to too smooth deformations. With decreasing λ, $\mathcal{E}_D$ decreases, while $\mathcal{E}_S$ increases. However, if λ gets very small, $\mathcal{E}_D$ can increase again, as the optimization problem gets more and more ill-posed.

Analyzing the L-curve for each warp estimation task is time-consuming and not always possible. Also, it is difficult to draw a general conclusion on the size of λ because it depends on the data itself (image and mesh resolution, gradient information, noise, type of motion etc.). Hence, the following strategy is pursued. For offline registration and warping tasks, different values for λ are iterated through on each pyramid level, from larger values to smaller values. In each iteration, the optimization is initialized with the results of the previous iteration. On the next pyramid level, the regularization parameter, for which the previous level converged, is scaled by 0.5, and an iteration through different values for λ starts again. For (online) tracking purposes, it is a good strategy to use 3-4 levels of resolution and to half the regularization parameter to the smoothness term on each resolution level and thereby relax the smoothness constraint on higher resolution levels.

Fig. 3.12 shows the influence of the regularization parameter λ_p on the smoothness of the photometric warp, illustrated as a shading map. The source and target

Figure 3.12.: Influence of the regularization parameter on the shading map. Left: source and target images. Right: shading maps generated from the estimated photometric parameters with $\lambda_p = 10, 1, 0.1, 0.01$ (from left to right, from top to bottom).

images (images of a pose-dependent clothing database, see Chapter 4) are depicted on the left. The source and target images do not only differ spatially because of articulated motion between them, but also the intensities change locally because of complex wrinkling patterns (e.g. at the arms) or cast shadows (e.g. at the torso). For large values of λ_p, the shading map generated from the photometric parameters is very smooth, and small details, such as fine wrinkling patterns, are not captured. With decreasing λ_p, more and more detailed shading patterns become visible in the shading map. However, if λ_p is chosen too small, the shading map can get noisy or texture details can get visible. This happens if differences due to spatial transformation are reduced by adapting the photometric warp parameters instead of the spatial parameters during optimization, i.e. if the regularization of the spatial and the photometric warp is unbalanced.

3.4. Chapter Summary

This chapter has introduced a framework for the joint estimation of spatial and photometric warps between two images. The warps are parameterized by mesh-based models, allowing a compact extraction of local texture deformation and shading differences between the images. The optimization is performed by minimizing a cost function with robust Quasi-Newton methods. The data term of the cost function is based on a relaxed brightness constancy assumption, which adapts the commonly used brightness constancy assumption to better reflect true conditions of changing and complex lighting. Thereby, intensity differences between corresponding image points are not handled as outliers but explicitly modeled in the cost function and warp model. A smoothness term exploits the mesh Laplacian to enforce local

smoothness of the deformation as well as the shading field. The results show that the photometric warp component not only allows the extraction of local shading differences between images for further exploitation but can also improve spatial registration and tracking based on intensity information. The approaches presented in the following chapters are based on the analysis of real images and exploit spatial and photometric warps extracted between them.

4. Pose-Space Image-Based Rendering

The goal of this thesis is to develop methods for a realistic visualization of clothing. Clothing that roughly follows a person's shape typically exhibits fine wrinkles and buckling patterns, especially near joints. For such type of clothing, it is a feasible assumption that wrinkling is pose-dependent [WHRO10]. Traditionally, the visualization of clothes in computer graphics relies on textured 3D models, and foldings, dynamics and reflection of the clothes are simulated with physics-based methods. Current cloth simulation techniques can produce highly realistic results for animated clothing with detailed wrinkling patterns but require complex and computationally demanding physical modeling of the cloth characteristics [BMF03, LYW+10, WHRO10]. For real-time applications, e.g. augmented reality environments, high-accurate cloth simulation methods are not applicable.

An alternative approach to physical simulation is observation of appearance through a number of images. In image-based rendering approaches, a database of previously recorded images is used to generate new images by clever interpolation and merging. Usually, the database contains view-dependent images of rigid objects, and the synthesis is restricted to viewpoint change. In contrast to that, this chapter presents a pose-dependent image-based rendering approach, which synthesizes images of a piece of clothing dependent on the articulated pose of a human body. The proposed method assumes that clothing appearance is pose- and view-dependent, concentrating on clothes that roughly fit the shape of a human body. New images for new pose configurations are interpolated and merged from a database of images, providing *examples* of the clothing's appearance for different body poses. This approach exploits the fact that all pose-dependent characteristics, such as texture deformation and shading properties, are implicitly captured by the example images. This information can be extracted from the database images as spatial and photometric warps between them and exploited to synthesize images of new pose configurations. The pose-dependency assumption allows a mapping of the extracted information onto an appropriate space, the *pose-space*, i.e. the space of body poses, to synthesize new images as a function of body pose [HE12, HFE13]. For the interpolation of image warps and intensities, scattered data interpolation methods, which have already been successfully used in example-based animation [LCF00, SRC01, ACP02, WHRO10], are exploited.

The proposed approach is mainly driven by the following assumptions:

- The aim is a plausible photo-realistic and perceptually correct visualization of clothes rather than accurate and correct reconstruction.
- Texture and shading represent strong cues for the perception of shape such that fine details can be modeled by images, carrying information on texture distortion, shading and silhouette (called *appearance*), whereas rough shape is modeled in a geometric model to allow animation.
- Wrinkles, creases and appearance of clothes that roughly follow a person's shape are mainly influenced by a person's pose. Although external forces and cloth dynamics can affect wrinkling behavior, appearance and shape of fine wrinkles are predominately affected by the joint angles of nearby body parts [WHRO10]. Under this assumption, the appearance of tight-fitting clothing can be modeled as a function of body pose.
- Wrinkling behavior is preliminary affected by the nearest joints. For example, the pose of the left elbow does not influence the wrinkling pattern on the right arm. This assumption allows a partition of the human body into several parts with lower degrees of freedom. This partitions the pose-space into subspaces, thereby reducing the dimensionality of the interpolation domain.

This chapter is structured as follows. Sec. 4.1 describes the proposed representation as well as the concept of pose-space parameterization in detail, before Sec. 4.2 focuses on the synthesis of new images from the database by scattered data interpolation in pose-space. Sec. 4.3 discusses experiments and results.

4.1. Pose-Dependent Image-Based Representation of Clothes

The main idea of the approach presented in this chapter is to capture a database of images from different calibrated viewpoints showing various body poses in an a-priori training phase. The poses, parameterized e.g. as a vector of skeleton joint angles or locations, position the images in pose-space, thereby providing examples of pose-dependent appearance at scattered points in this space. Input to the synthesis step (Sec. 4.2) is a new pose configuration, defining the position in pose-space to be interpolated from the database. Sec. 4.1.1 describes the definition of the image-based representation in pose-space before Sec. 4.1.2 focuses on the construction of pose-dependent databases in practice.

4.1.1. Database Definition

This section concentrates on the definition of the database representation. Details on the construction of a database, such as depth map estimation, pose recovery, etc. are given in Sec. 4.1.2. A pose-dependent database (Fig. 4.1) consists of a set of

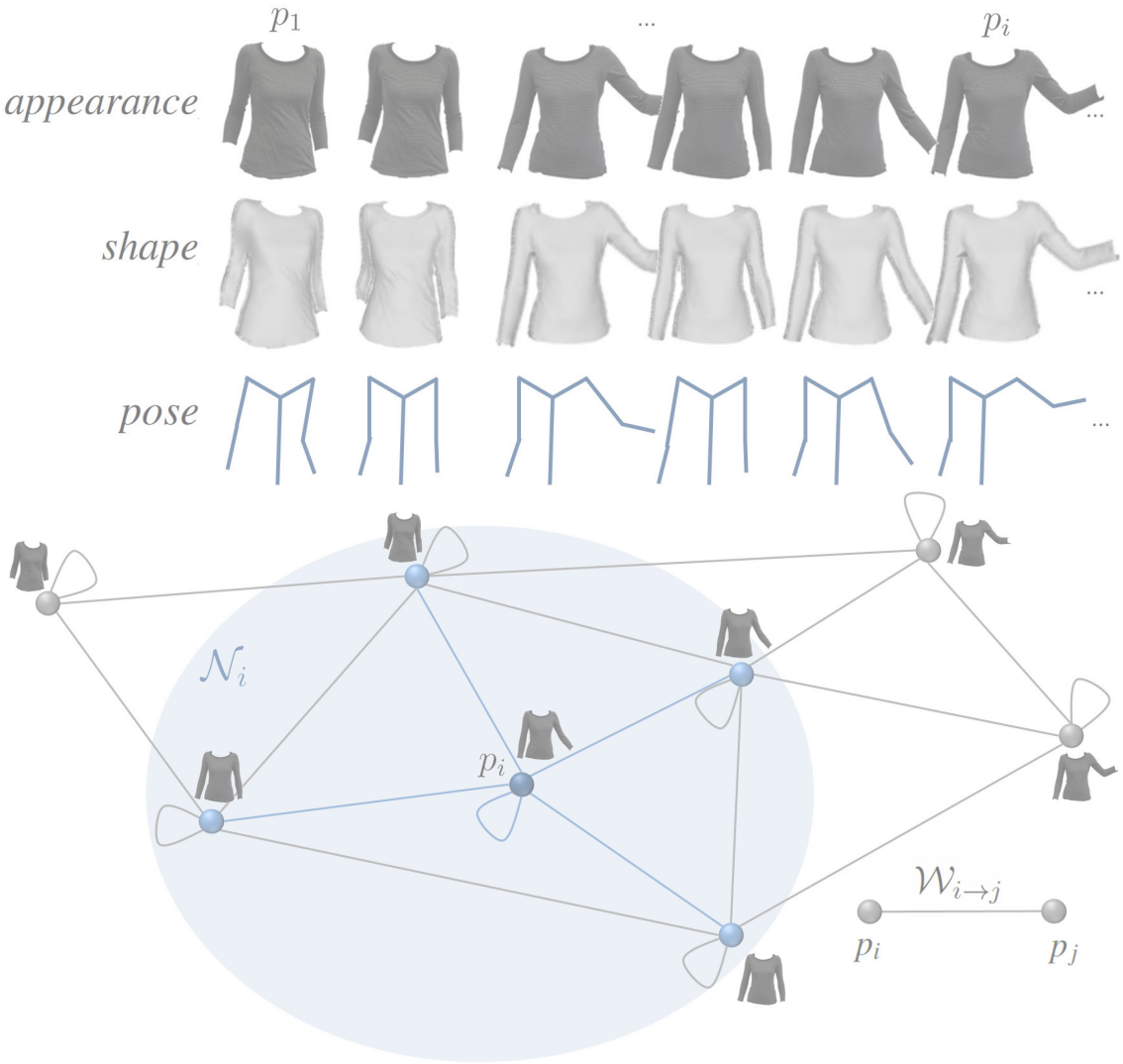


Figure 4.1.: This chapter introduces a new approach to image-based rendering in pose-space. The key concept is to capture examples of clothing appearances, i.e. images of clothes, in various poses. Each image is associated with a skeleton model, parameterizing the images at scattered positions in pose-space. Additionally, the database stores parameters for spatial and photometric image warps between subsets of images that lie close to each other in pose-space. This information is stored in a pose-graph, where nodes represent poses and an edge between two poses p_i and p_j represents a known image warp between the images $\mathcal{I}_i$ and $\mathcal{I}_j$. The neighborhood $\mathcal{N}_i$ of a pose p_i in the pose-graph is defined by all poses $p_n \in \mathcal{N}_i$, which are connected to p_i by an edge in the pose-graph, i.e. all poses where an image warp from p_i is known to.

calibrated images $\mathcal{I}_{p,v}(\mathbf{x})$ (appearance) showing various body poses $p = 1 \dots P$ captured from different calibrated viewpoints $v = 1 \dots V$. Each image is associated with an alpha mask $\mathcal{A}_{p,v}(\mathbf{x}) \in [0, 1]$ and a view-dependent 3D mesh $\mathcal{M}^3_{p,v} : \{\mathbf{V}^3_{p,v}, \mathcal{F}_{p,v}\}$, representing a depth map (shape). For animation and parameterization purposes, each image is associated with a skeleton, representing the body pose, parameterized in a vector $\mathbf{q}_p$, e.g. a vector of joint angles or locations, which positions the image

in pose-space (Fig. 4.2). For animation of the mesh, skinning weights between the mesh and the skeleton are stored in the database.

The images contain information on complex pose-dependent appearance, i.e. texture deformation and shading. This information can be implicitly extracted from the images as warps between them. For this purpose, image warps (spatial and photometric warps) are estimated between subsets of images that lie close to each other in pose-space. The parameters for these warps are stored in the database. The information, which images are connected by warps, is organized in a graph structure, called *pose-graph*, in which nodes represent database images and edges represent warps between them (Fig. 4.1).

This section explains the concept of the pose-space and pose-graphs as well as mesh-based spatial and photometric database warps. For simplicity reasons, the differentiation between poses p and views v will be omitted in the following. In fact, different viewpoints can be treated as different poses such that interpolation between viewpoints is akin to the presented pose interpolation. In the following, p_i and p_j are used to indicate different pose/view-entries in the database, the associated images are denoted by $\mathcal{I}_i, \mathcal{I}_j$, and a warp between $\mathcal{I}_i$ and $\mathcal{I}_j$ is denoted by $\mathcal{W}_{i\to j}$. Meshes, alpha masks, pose vectors etc. are denoted with according subscripts.

4.1.1.1. Pose-Space and Pose-Graph

Pose-Space. The proposed image-based representation is pose-dependent and the space of human body poses is used as a domain for interpolation. To represent a human body pose and to allow animation, a representation and parameterization of a body pose is needed. Usually, a human body pose is represented by a skeleton of bones connected by articulated joints, structured in a tree-topology (Fig. 4.2). One joint is selected as the root, and the other joints are connected up in a tree hierarchy. The joints allow relative movements of the bones within the skeleton, represented by local transformation matrices. Depending on the joint type, each joint has one or more degrees of freedom, defining its possible range of motion. For example, a hinge joint, such as an elbow, has one degree of freedom, and a ball-in-socket joint, such as a shoulder, has 3 degrees of freedom. These values represent a specific pose (position and orientation of each joint), and a local matrix can be constructed for each joint, specifying its orientation (or the orientation of its adjacent bone) to the joint above it in tree hierarchy. In general, this 4×4 local transformation matrix represents a rigid body transformation (in homogeneous coordinates) as

$$\mathbf{T}_{joint} = \begin{bmatrix} \mathbf{R}_{joint} & \mathbf{t}_{joint} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} ,$$

where $joint = 1 \dots J$ denotes the joint index. The translational part $\mathbf{t}_{joint}$ is fully defined by the bone lengths and the orientation of the parent joint such that the degrees of freedom are part of the rotation matrix $\mathbf{R}_{joint}$ only. The local transformation matrices are defined in the parent joint's local coordinate frame and can now

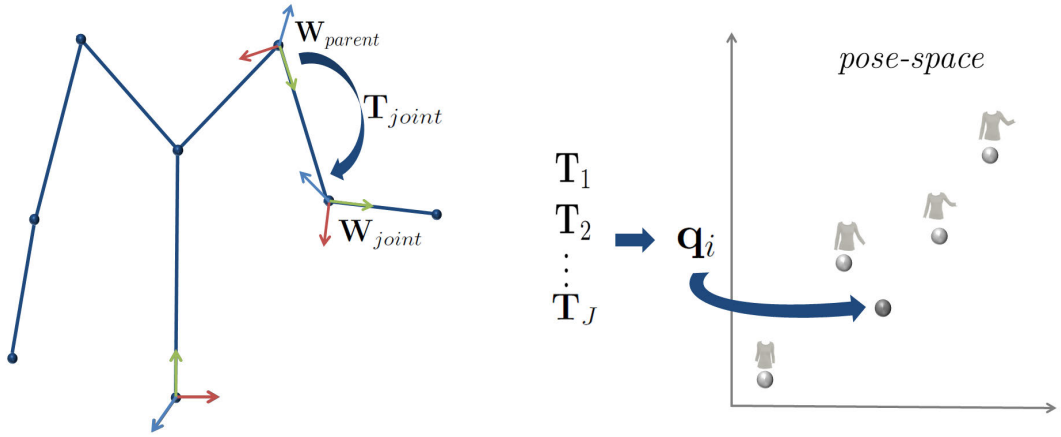


Figure 4.2.: A pose p_i is typically represented by a skeleton of bones connected by joints. Each joint is characterized by a 4×4 matrix $\mathbf{T}$, specifying its position and orientation with respect to its parent joint. A pose parameterization $\mathbf{q}_i$ deduced from these matrices, e.g. a vector of joint angles or locations, positions the pose in pose-space.

be used to compute a *world matrix* $\mathbf{W}_{joint}$ for each joint, specifying its position and orientation in the world coordinate system:

$$\mathbf{W}_{joint} = \mathbf{T}_{joint} \cdot \mathbf{W}_{parent} \ .$$

$\mathbf{W}_{parent}$ denotes the world matrix of the parent joint. With these matrices, the skeleton is fully described, and any parameterization $\mathbf{q}_i$ of a human body pose p_i will make use of these matrices (Fig. 4.2).

The parameterization of a pose in a vector $\mathbf{q}$, and accordingly a definition of the pose-space as well as a distance or similarity measure in this space, is not trivial [CZN+11]. One possible representation is a *vector of joint angles*. The angle representation can be a set of Euler angles [HG09], axis-angle representation [SFBH09] or quaternions [WSLG07], derived from the local matrices $\mathbf{T}_{joint}$. For joints with only one degree of freedom, such as the elbows, only one angle is stored in the parameter vector. A different approach is to define a pose and a similarity measure by exploiting joint locations in a normalized coordinate system [XLS+11]. Given a vector of such parameterized joint angles or joint locations, a simple distance measure between two poses in pose-space is the L^2-norm of the difference of two pose vectors:

$$d_{ij}^{L^2} = d^{L^2}(\mathbf{q}_i, \mathbf{q}_j) = \|\mathbf{q}_i - \mathbf{q}_j\| \ . \tag{4.1}$$

The pose parameterization $\mathbf{q}$ spans the space of all possible pose configurations, i.e. the pose-space, which is high dimensional. This high dimensionality will be handled by defining suitable subspaces of different parts of the body that can be handled more or less independently (Sec 4.2.3). This approach assumes, similar to Allen et al. [ACP02] as well as Wang et al. [WHRO10], that joints only influence nearby bones.

Pose-Graph. Each image $\mathcal{I}_i$ is now associated with a pose parameterization $\mathbf{q}_i$, which defines its position in pose-space. Image warps are estimated between *close*

images in pose-space, i.e. image pairs with a distance $d_{ij}^{L^2}$ below a predefined threshold (Equ. (4.1)). A pose-graph stores the information which images are connected by warps. In this graph, nodes represent all database entries, and edges represent image warps. Each entry is also connected to itself, as the transformation to itself is known (*identity warp*) and will be needed for pose-space interpolation (compare Equ. (4.10)-(4.12)). The neighborhood of a pose p_i, i.e. all poses that are connected to p_i by an edge, is defined by $\mathcal{N}_i$ (Fig. 4.1 on page 57). Hence, $\mathcal{I}_n, p_n \in \mathcal{N}_i$ defines the set of images with associated warps from and to $\mathcal{I}_i$.

4.1.1.2. Pose-Space Warps

For each edge in the pose-graph, joint spatial and photometric image warps as defined in Chapter 3 are estimated and stored in the database. For later interpolation, the warps between database images are split up into a coarse image warp, animating an image to a new pose, and a subsequent detail-warp, adding fine pose-dependent and contextual details. For the coarse warp, standard animation techniques can be used, which move the mesh vertices dependent on the animation of the underlying skeleton bones, e.g. skeletal subspace deformation (SSD) with linear blend skinning (LBS) [LCF00]. Such a warp is fully defined by the mesh geometry and the skeleton configuration. Fine contextual and pose-dependent details, not captured by the SSD-warp, are added by the detail-warp. The idea behind this partition is that during rendering, the SSD-warp can be used to coarsely animate a database image to any new pose configuration, and fine pose-dependent details can be interpolated from the detail-warps stored in the database. In this section, details on the database warp models are given.

To animate an image $\mathcal{I}_i$ to another database pose p_j, the associated 3D mesh $\mathcal{M}_i^3$ is animated by skeletal subspace deformation (SSD) with linear blend skinning (LBS) and projected into the desired camera view. SSD with LBS moves each vertex of the mesh by a weighted linear blend of bone transformations (Equ. (2.1) on page 16). The such established 2D vertex correspondences are then used to warp the image to the new pose and view. These coarse warps, induced by SSD-animation and projection, will be called *SSD-warps* in the following and are defined by

$$\mathcal{W}_{i\to j}^{\mathcal{SSD}} : \{\mathbf{V}_i, \mathcal{F}_i, \Delta\mathbf{V}_{i\to j}^{\mathcal{SSD}}, \mathbf{1}, \} \ , \tag{4.2}$$

with

$$\begin{aligned} \mathbf{V}_i &= \mathcal{P}_{\mathbf{P}_i}(\mathbf{V}_i^3) \\ \Delta\mathbf{V}_{i\to j}^{\mathcal{SSD}} &= \mathcal{P}_{\mathbf{P}_j}\big(\mathcal{SSD}_{i\to j}(\mathbf{V}_i^3)\big) - \mathcal{P}_{\mathbf{P}_i}(\mathbf{V}_i^3) \ . \end{aligned} \tag{4.3}$$

$\mathbf{V}_i$ denotes the original vertices of $\mathcal{M}_i^3$ projected into the camera of $\mathcal{I}_i$, where $\mathcal{P}_{\mathbf{P}}(\mathbf{V}^3)$ describes the projection of the 3D mesh vertices $\mathbf{V}^3$ with the camera projection matrix $\mathbf{P}$ (see App. A.1.1). $\Delta\mathbf{V}_{i\to j}^{\mathcal{SSD}}$ denotes the vertex displacements induced by SSD animation $\mathcal{SSD}_{i\to j}$ from pose p_i to p_j, according to Equ. (2.1) on page 16, and projection into the desired view $\mathbf{P}_j$. The SSD-warp is a purely spatial warp (the

intensity scale for all vertices is 1) but denoted as a joint warp as it will later be concatenated with joint spatial and photometric detail-warps. This warp is fully defined by the geometry and pose information associated to each image. Hence, no warp parameters need to be stored.

The SSD-warps coarsely animate an image to a new pose, but they do not register the images accurately. This is because of the following reasons. First, SSD does not model real contextual deformation at joints, e.g. cloth wrinkling or muscle bulging when an arm is bent. On the contrary, SSD is even known to introduce artifacts near the joints, e.g. the collapsing joint defect [LCF00] where the skin near joints collapses for increasing bending and twisting angles. Second, the depth maps might be inaccurate such that the images are not registered correctly. Lastly, a change in pose not only changes the shape of the clothes but also texture and shading. To compensate for the inaccuracies in the SSD-warp and to assure photo-consistency between poses, additional (spatial and photometric) detail-warps are extracted on top of the SSD-warps:

$$\mathcal{W}^{\mathcal{D}}_{i\to n} : \{\mathbf{V}_i + \Delta\mathbf{V}^{\mathcal{SSD}}_{i\to n}, \mathcal{F}_i, \Delta\mathbf{V}^{\mathcal{D}}_{i\to n}, \boldsymbol{\rho}^{\mathcal{D}}_{i\to n}\} \ . \tag{4.4}$$

These warps are extracted between subsets of database images $\mathcal{I}_i$ and $\mathcal{I}_n, p_n \in \mathcal{N}_i$, which lie close to each other in pose-space. These images are connected by an edge in the pose-graph. The concatenated SSD- and detail-warps then register the images accurately both spatially and photometrically (compare Fig. 4.6-4.7 on pages 65-66).

Formally, the concatenation $\oplus$ of warps between two database images $\mathcal{I}_i$ and $\mathcal{I}_n, p_n \in \mathcal{N}_i$ can be expressed as

$$\begin{aligned} \mathcal{W}_{i\to n} &: \mathcal{W}^{\mathcal{SSD}}_{i\to n} \oplus \mathcal{W}^{\mathcal{D}}_{i\to n} \\ &: \{\mathbf{V}_i, \mathcal{F}_i, \Delta\mathbf{V}^{\mathcal{SSD}}_{i\to n} + \Delta\mathbf{V}^{\mathcal{D}}_{i\to n}, \boldsymbol{\rho}^{\mathcal{D}}_{i\to n}\} \ . \end{aligned} \tag{4.5}$$

The parameters $\Delta\mathbf{V}^{\mathcal{D}}_{i\to n}$ and $\boldsymbol{\rho}^{\mathcal{D}}_{i\to n}$ of the detail-warps are stored in the database and will be interpolated in pose-space during rendering (Sec. 4.2.1).

4.1.1.3. Summary

To conclude, for each pose p_i, the database consists of

- A coarse 3D mesh-based depth map $\mathcal{M}^3_i : \{\mathbf{V}^3_i, \mathcal{F}_i\}$ together with a skeleton model, representing the body pose in a vector $\mathbf{q}_i$, allowing for dominant motions, e.g. view interpolation and coarse animation (SSD-warp).
- An image $\mathcal{I}_i$, which can be regarded as pose-dependent appearance example of the pose p_i. At the silhouette, fine details are accounted for through an alpha-mask $\mathcal{A}_i \in [0\ 1]$ associated to the captured images.

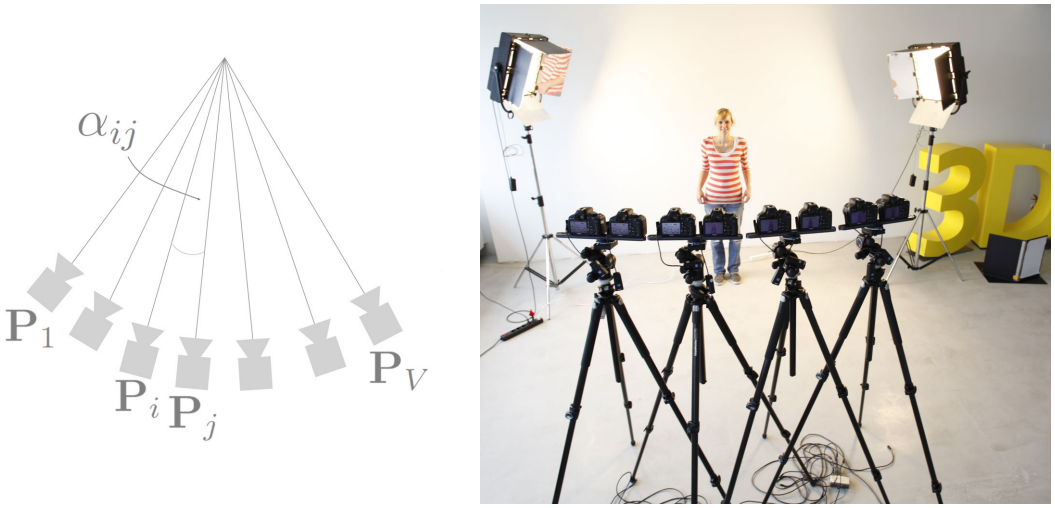


Figure 4.3.: Typical camera setup for the acquisition of a multi-view multi-pose database. Camera distances are measured as angular difference between the cameras' line of sight.

- Spatial and photometric warps to other database poses $p_n \in \mathcal{N}_i$, which lie close to p_i in pose-space. These poses are connected by an edge in the pose-graph. The warps are separated into a coarse SSD-warp $\mathcal{W}^{\mathcal{SSD}}_{i\to n}$, which coarsely animates an image to a new pose, and a detail-warp $\mathcal{W}^{\mathcal{D}}_{i\to n}, p_n \in \mathcal{N}_i$, which adds fine pose-dependent details not captured in the SSD-warp. The parameters of the detail-warp $\Delta\mathbf{V}^{\mathcal{D}}_{i\to n}, \boldsymbol{\rho}^{\mathcal{D}}_{i\to n}$ are stored in the database.

4.1.2. Database Generation

Input data to the database creation is a set of multi-view images $\mathcal{I}_{v,p}$ taken from different viewpoints $v = 1 \dots V$ and showing different body poses $p = 1 \dots P$ (Fig. 4.3). Each image is segmented to an alpha mask $\mathcal{A}_{v,p} \in [0\ 1]$. The generation of the database proceeds in the following steps:

Calibration and Depth Map Estimation. For calibration and reconstruction of the coarse geometry, the bundle adjustment method of [SSS08] is exploited. This method takes a multi-view image set (of the same pose) as input and outputs a camera matrix $\mathbf{P}_v = \mathbf{K}_v\,[\mathbf{R}_v\ \mathbf{t}_v]$ (see App. A.1.1) for each view, as well as sparse 3D points $\mathbf{b}^3_i, i = 1 \dots N$. To generate a mesh-based depth map for each view and pose, the sparse reconstructions are refined by registering two neighboring stereo pairs of the same pose onto each other using the mesh-based image warp optimization method of Chapter 3 (Fig. 4.4).

To determine the stereo pairs of *neighboring* cameras, for each (source) view $v_\mathcal{S}$, the nearest (target) view $v_\mathcal{T}$ is determined that minimizes the viewing angle between the cameras (Fig. 4.3). To refine the correspondences between the stereo images, a 2D mesh $\mathcal{M}_\mathcal{S} : \{\mathbf{V}, \mathcal{F}\}$ is generated on the source image by triangulating the alpha mask using the method of [She96]. Exploiting the warp optimization method presented in Chapter 3, mesh-based image warps are estimated between the two

Figure 4.4.: From left to right: example sparse point cloud resulting from the bundle adjustment method of [SSS08]; 3D mesh generated from these points; refined 3D mesh after image-based warp optimization.

images. The warp optimization is initialized with sparse correspondences induced by the 3D points $\mathbf{b}_i^3$ projected into the source and the target images:

$$\begin{aligned} \mathbf{b}_{\mathcal{S}i} &= \mathcal{P}_{\mathbf{P}_\mathcal{S}}(\mathbf{b}_i^3) \\ \mathbf{b}_{\mathcal{T}i} &= \mathcal{P}_{\mathbf{P}_\mathcal{T}}(\mathbf{b}_i^3) \ . \end{aligned} \tag{4.6}$$

Here, $\mathcal{P}_\mathbf{P}(\mathbf{b}^3)$ describes the projection of the 3D point $\mathbf{b}^3$ with the camera projection matrix $\mathbf{P}$ (see App. A.1.1), and $\mathbf{P}_\mathcal{S}$, $\mathbf{P}_\mathcal{T}$ denote the camera matrices of the source and the target views. Warp initialization with sparse correspondences is detailed in Sec. 3.2.4. From the estimated vertex correspondences and calibration information, the 3D position of each mesh vertex is determined and stored in a depth map, represented as a 3D mesh $\mathcal{M}_{p,v}^3 : \{\mathbf{V}_{p,v}^3, \mathcal{F}_{p,v}\}$ for each image $\mathcal{I}_{p,v}$ [HZ04]. Fig. 4.4 compares mesh-based depth maps reconstructed from the sparse correspondences [SSS08] and after mesh-based warp refinement.

Skeleton Attachment. To recover the human body pose for each multi-view image set, a generic body model associated with skeleton information is fitted to the mesh-based depth maps of each multi-view image set of the same pose exploiting the method described in [FHE12] with visual inspection and possible manual corrections. As the depth maps have different topologies, the vertices and vertex normals of the same pose but different viewpoints are consolidated into one oriented point cloud used as target for template fitting. The fitted skeleton is assigned to all images of the same pose p.

The skeletons and the meshes are disconnected until skinning weights specify how to apply skeleton bone transformations to the vertices. For animation, skeleton subspace deformation (SSD) is used, which transforms each vertex of a skinned mesh by a weighted linear blend of bone transformations, according to Equ. (2.1) on page 16. The weights determine the influence of each bone on each vertex, and are calculated with the method of [BP07], which solves for a heat equilibrium over the mesh surface to generate a smooth weighting field.

From the skeletons, a pose parameterization vector $\mathbf{q}$, representing the example's position in pose-space, is extracted. Two different pose parameterizations have been used: a representation based on joint angles, and a representation based on relative

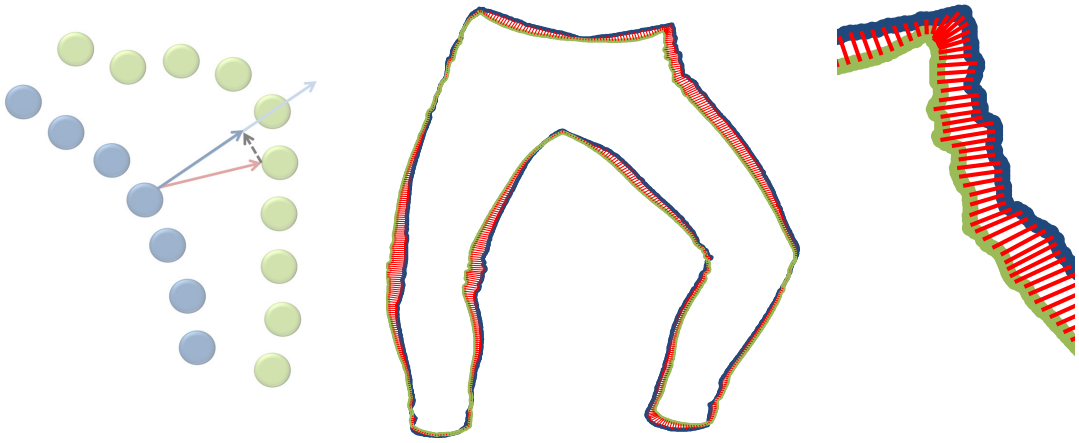


Figure 4.5.: Left: illustration of the silhouette ICP. Source (blue) and target (green) silhouettes are sampled, and for each source point the nearest target point is found (red arrow). The vector between corresponding points is projected onto the source normal (light blue) to yield the displacement vector (dark blue) for each iteration. Right: established correspondences after 5 iterations.

joint locations. For the angle-based representation, an axis-angle representation of the joint rotation is used, and the axis is represented by its azimuth and elevation angles. For the pose representation based on joint locations, the locations are measured in the coordinate system of the torso joint.

Detail-Warp Optimization. Using the pose data with the skinned meshes as well as calibration information, an SSD-warp (Equ. (4.2)) can now coarsely warp one pose image onto any other database pose. However, the SSD-warps do not model real deformations and texture modifications that occur when pose is changed. For this reason, additional detail-warps are needed, capturing those characteristics. Such warps

$$\mathcal{W}^{\mathcal{D}}_{i\to n} : \{\mathbf{V}_i + \Delta\mathbf{V}^{\mathcal{SSD}}_{i\to n}, \mathcal{F}_i, \Delta\mathbf{V}^{\mathcal{D}}_{i\to n}, \boldsymbol{\rho}^{\mathcal{D}}_{i\to n}\}$$

are estimated on top of the SSD-warps between poses p_i and $p_n \in \mathcal{N}_i$ that lie close to each other in the pose-space and are connected by an edge in the pose-graph, as detailed in the following.

Starting from the SSD-warped image $\mathcal{W}^{\mathcal{SSD}}_{i\to n}\big(\mathcal{I}_i(\mathbf{x})\big)$, an additional mesh-based warp is estimated that fine-scale aligns the two database images both in the spatial as well as in the intensity domain if concatenated as in Equ. (4.5) (Fig. 4.6-4.7). The warp optimization is performed using the method described in Chapter 3. Sparse SIFT [Low03] as well as silhouette correspondences are determined to assure a good initialization (Sec. 3.2.4). The SSD-warped image and the target pose image are already coarsely registered, and the difference is most obvious at the silhouettes as SSD-animation does not capture the contextual deformation of clothes and muscles (Fig. 4.5). The SIFT correspondences account for correspondences on the texture, whereas the silhouette correspondences support an accurate registration of the silhouettes. To determine the silhouette correspondences, the silhouettes of the SSD-warped image and the target pose image are registered in a non-rigid iterative

Figure 4.6.: Top, from left to right: source and target poses overlaid; source pose image warped with the SSD-warp; source pose image warped with an additional detail-warp. Bottom: corresponding difference images. Note the adapted shading patterns in the image warped with the concatenated SSD- and detail-warp (see also Fig. 4.7).

closest point (ICP) approach [BM92] (Fig. 4.5). Let $\mathbf{s}_{\mathcal{S}i}$ and $\mathbf{s}_{\mathcal{T}i}, i = 1 \dots N$ denote the sampled silhouette points in the SSD-warped image (source image) and the target pose image. For each point in the source point set, the nearest point in the target point set (and vice versa) is determined. Matches found in both matching directions are selected, and point pairs with a distance above a predefined threshold are discarded. As this threshold depends on the data, it is determined using MAD-based outlier rejection (App. A.1.2). For the remaining points, the displacement vectors from the source points to the matched target points are projected onto the source point normals to allow only displacements $\mathbf{d}$ along the point normals (Fig. 4.5). The silhouette point displacements $\Delta\mathbf{s}_{\mathcal{S}i}$ of the complete source point set are calculated with a Laplacian smoothness constraint on all silhouette points (compare App. A.1.3):

$$\begin{bmatrix} \mathbf{D} \\ \lambda\mathbf{L}_s \end{bmatrix} \Delta\mathbf{S} = \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad . \tag{4.7}$$

Here, $\mathbf{D} = [\mathbf{I}_n \ \mathbf{0}]$, and $\mathbf{d}_x$ and $\mathbf{d}_y$ are the concatenated column vectors of the displacements in $x-$ and $y-$direction. $\Delta\mathbf{S} = [\Delta\mathbf{s}_{\mathcal{S}1} \dots \Delta\mathbf{s}_{\mathcal{S}N}]^T$ is a matrix of all source point displacements, ordered such that for the first $n \leq N$ points displacements $\mathbf{d}$ have been calculated. The silhouette Laplacian $\mathbf{L}_s$ stores the neighborhood information of the silhouette point set and has the following entries:

$$l_{ij} = \begin{cases} -1 & i = j \\ \frac{1}{2} & \text{if points } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \quad . \tag{4.8}$$

The source point set is displaced with the estimated displacements. After each iteration, the mean distance between corresponding points is calculated, and the

Figure 4.7.: Influence of the photometric detail-warp on wrinkling appearance. From top to bottom/left to right: source image warped onto target pose without photometric warp; source image warped onto target pose with photometric warp; target pose image. Complex shading patterns can be modeled by the photometric warp.

iteration is stopped if the change in the mean distance is below a predefined threshold. Otherwise the procedure starts from the beginning with the displaced source point set.

Let $\mathbf{b}_{\mathcal{S}i}$ and $\mathbf{b}_{\mathcal{T}i}$ denote silhouette as well as SIFT correspondences in the source and in the target image. These sparse point correspondences are used to initialize the warp optimization approach of Chapter 3. Details on warp initialization with sparse correspondences can be found in Sec. 3.2.4. The optimized parameters of the detail-warp $\Delta\mathbf{V}^{\mathcal{D}}_{i\to n}, \boldsymbol{\rho}^{\mathcal{D}}_{i\to n}$ on top of the SSD-warp are stored in the database. These parameters capture fine pose-dependent details and will be interpolated during rendering. Fig. 4.6-4.7 illustrate the benefits of the fine-scale registration of pose images.

4.2. Pose-Space Image-Based Rendering

In the following pose-space image-based rendering approach (PS-IBR), an image for an arbitrary pose p_a is synthesized from the *closest* m images $\mathcal{I}_i$ to p_a in pose-space. This section is structured as follows. Sec. 4.2.1 describes how warps to the new pose p_a are interpolated from the stored database warps for each selected image, before Sec. 4.2.2 focuses on the blending of the warped images. Sec. 4.2.3 concentrates on a separation of the pose-space into subspaces to allow for a larger variety of synthesized poses and to reduce the number of required examples. Sec. 4.2.4 analyses appropriate distance measures for the selection of these *closest* m images in pose-space. Finally,

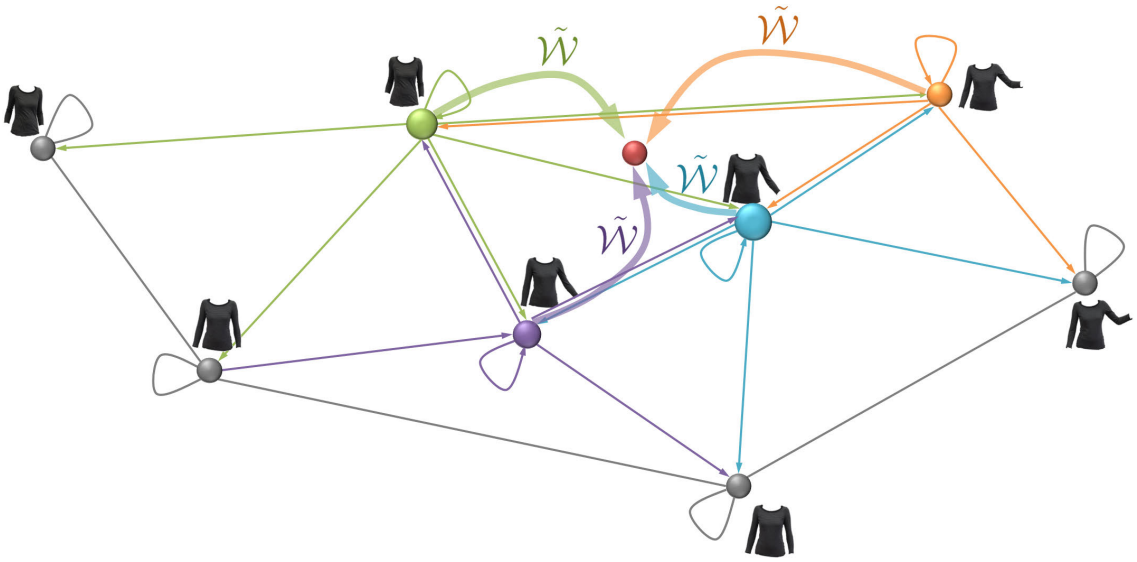


Figure 4.8.: Illustration of pose-space image-based rendering. Each image in the database is positioned in pose-space by a skeleton parameterization (graph nodes). For a new incoming pose (marked in red), the closest m images are selected (depicted in different colors; the dot sizes illustrate the blending weights). For each selected image, a warp to the new position in pose-space is unknown and interpolated from the stored database warps (graph edges/colored arrows).

Sec. 4.2.5 presents details on how joint changes in pose and viewpoint are handled.

4.2.1. Scattered Warp Interpolation

Suppose, the m *closest* pose images to the new pose p_a have already been selected to synthesize the new image (an analysis of a suitable distance measure is presented in Sec. 4.2.4). For each of the selected images $\mathcal{I}_i$, a warp to the pose p_a is unknown and needs to be interpolated from the stored image warps $\mathcal{W}_{i\to n}, p_n \in \mathcal{N}_i$. Recall that warps between database images are concatenated SSD- and detail-warps. The SSD-warps are fully defined by the skinning weights and the skeleton information. Hence, only the detail-warp parameters added to the SSD-warp need to be interpolated from the stored parameters. The idea behind partitioning the warps into an SSD-warp and a detail-warp is that the SSD-warp can coarsely animate each database image to a new pose in pose-space, and fine details, not captured by the SSD-warp, can be interpolated from the stored database warps. Let $\tilde{\mathcal{W}}_{i\to a}$ denote the interpolated warp from a database pose p_i to an arbitrary pose p_a:

$$\tilde{\mathcal{W}}_{i\to a} = \mathcal{W}^{\mathcal{SSD}}_{i\to a} \oplus \tilde{\mathcal{W}}^{\mathcal{D}}_{i\to a} \ . \tag{4.9}$$

The interpolated detail-warp $\tilde{\mathcal{W}}^{\mathcal{D}}_{i\to a}$ is represented by the interpolated vertex displacements $\Delta\tilde{\mathbf{V}}^{\mathcal{D}}_{i\to a}$ and intensity scale parameters $\tilde{\boldsymbol{\rho}}^{\mathcal{D}}_{i\to a}$.

The poses $p_n \in \mathcal{N}_i$ define the set of example poses to which a warp from p_i is known to. These poses are located at scattered positions in pose-space, and the interpolation problem is thus a problem of scattered data interpolation. One interpolation strategy, which is also used in some PSD approaches [ACP02, SRC01], is to find a smooth weight function per example such that the interpolated warp parameters are a linear combination of the stored warp parameters:

$$\begin{aligned} \Delta\tilde{\mathbf{V}}^{\mathcal{D}}_{i\to a} &= \sum_{p_n\in\mathcal{N}_i} w_n(\mathbf{q}_a)\cdot\Delta\mathbf{V}^{\mathcal{D}}_{i\to n} \\ \tilde{\boldsymbol{\rho}}^{\mathcal{D}}_{i\to a} &= \sum_{p_n\in\mathcal{N}_i} w_n(\mathbf{q}_a)\cdot\boldsymbol{\rho}^{\mathcal{D}}_{i\to n} \quad . \end{aligned} \tag{4.10}$$

This means that weight functions $w_n(\mathbf{q})$ need to be defined for all poses $p_n \in \mathcal{N}_i$ as a function of $\mathbf{q}$. For a smooth and plausible interpolation of warps, the weight functions should meet four constraints:

- The weights should sum to one at the new position in pose-space:

$$\sum_{p_n\in\mathcal{N}_i} w_n(\mathbf{q}_a) = 1 \ . \tag{4.11}$$

- If p_a falls onto a pose $p_n \in \mathcal{N}_i$, i.e. $\mathbf{q}_a = \mathbf{q}_n$, the weight for that sample must be one, and all other weights must be zero:

$$\begin{aligned} w_n(\mathbf{q}_a) &= 1 \ \text{ if } \mathbf{q}_n = \mathbf{q}_a \\ w_n(\mathbf{q}_a) &= 0 \ \text{ if } \mathbf{q}_n \neq \mathbf{q}_a \ . \end{aligned} \tag{4.12}$$

- The weight functions should be smooth and continuous so that the transitions are natural during animation.
- To ensure reliable interpolated warp parameters, the weight functions should be non-negative.

In the PSD literature, local deterministic scattered data interpolation methods, such as radial basis functions (RBF) [SRC01, LCF00] or k-nearest neighbors (kNN) interpolation [ACP02] have been proposed. These approaches use local information of nearby scattered data points to calculate a smooth interpolation function. However, radial basis functions can yield negative weight functions such that the fourth constraint is not met (compare Fig. 4.9-4.10). Negative weight functions can lead to unreliable and exaggerated warps. In contrast, the kNN-interpolation method fulfills all four constraints, including non-negativity (Fig. 4.9). For this reason, kNN interpolation is exploited in this thesis as detailed below. After the weight functions have been determined, a warp to the new pose p_a is interpolated for each selected image according to Equ. (4.10). With these interpolated warps $\tilde{\mathcal{W}}_{i\to a}$, the images are warped to the arbitrary pose p_a and blended as described in Sec. 4.2.2.

k-Nearest Neighbors Interpolation. K-nearest neighbors (kNN) interpolation has been proposed by Buehler et al. for view interpolation in image-based rendering

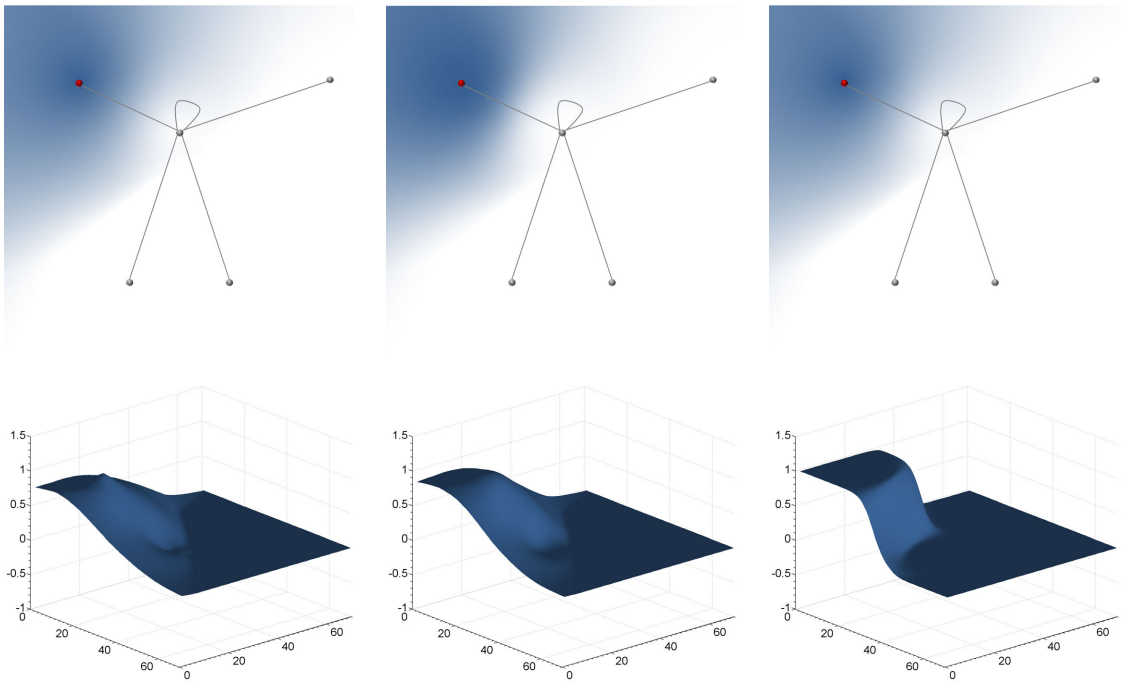


Figure 4.9.: Illustration of kNN-interpolation for the weight function $w_k(\mathbf{q})$ for one control point $\mathbf{q}_k, p_k \in \mathcal{N}_i$ (marked in red) in a 2D pose-space. The value of the weight function is interpolated from all known control points $\mathbf{q}_n, p_n \in \mathcal{N}_i$ ($\mathbf{q}_i$ is the data point in the center). It takes the value 1 at $\mathbf{q}_k$ and 0 at all other control points. From left to right: increasing values for κ ($\kappa = 1, 2, 10$). In the upper row, more saturated colors represent higher weights. Fig. 4.10 plots weight functions for the same example achieved with Gaussian RBF interpolation.

[BBM+01] and also used by Allen et al. in pose-space deformation [ACP02]. The idea goes back to inverse-distance weighting proposed by Shepard [She68] for multivariate scattered data interpolation. Shephard's method assigns each of the control points $\mathbf{q}_n$ a weight function

$$f_n(\mathbf{q}_a) = \frac{1}{d(\mathbf{q}_a, \mathbf{q}_n)^\kappa} \tag{4.13}$$

based on their inverse distance $d(\mathbf{q}_a, \mathbf{q}_n)$ to the new position $\mathbf{q}_a$ in pose-space. One possible choice for the distance measure is the L^2-norm $d(\mathbf{q}_i, \mathbf{q}_j) = d^{L^2}(\mathbf{q}_i, \mathbf{q}_j)$ (Equ. (4.1) on page 59). κ is an exponent controlling the shape of the interpolation function. With increasing values of κ, greater influence is assigned to control points close to the interpolated point, with the interpolated function turning into a mosaic of nearly constant values for very large values of κ.

In kNN-interpolation, the weight function is altered by

$$f_n(\mathbf{q}_a) = \begin{cases} \frac{1}{d(\mathbf{q}_a, \mathbf{q}_n)^\kappa} - \frac{1}{d(\mathbf{q}_a, \mathbf{q}_k)^\kappa} & \text{if } d(\mathbf{q}_a, \mathbf{q}_n) \leq d(\mathbf{q}_a, \mathbf{q}_k) \\ 0 & \text{else} \end{cases}, \tag{4.14}$$

where $\mathbf{q}_k$ is a pose parameterization of the k^{th} farthest pose $p_k \in \mathcal{N}_i$ to the new pose parameterization $\mathbf{q}_a$. Finally, the weights are normalized such that they sum

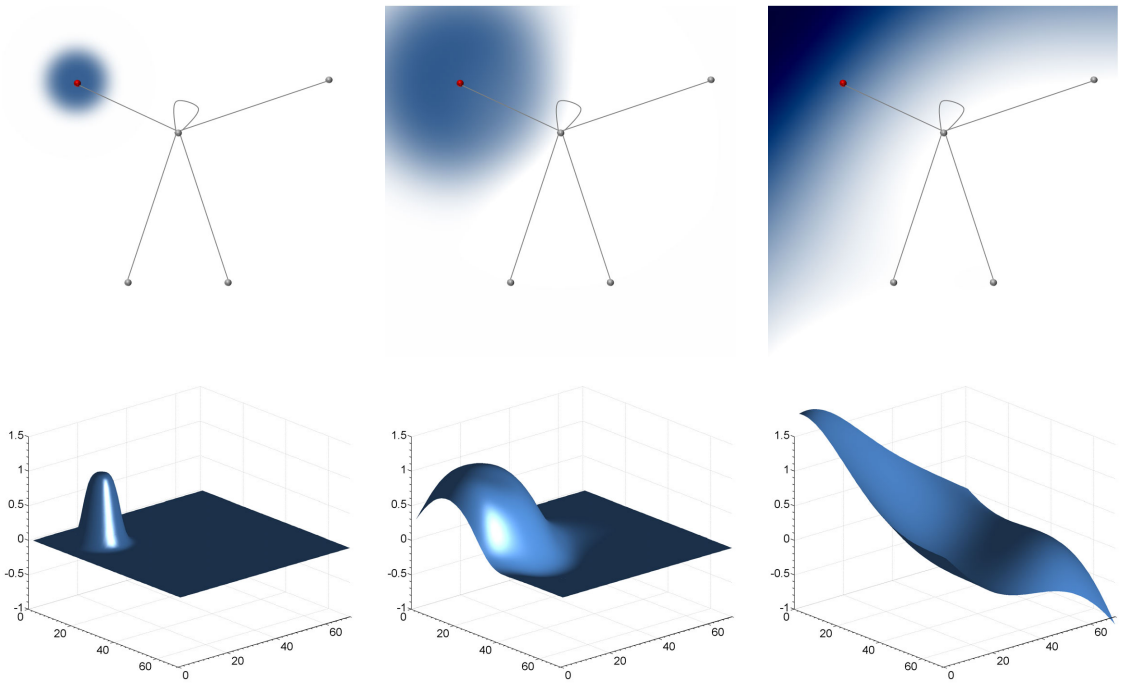


Figure 4.10.: Illustration of RBF interpolation for the weight function $w_k(\mathbf{q})$ for one control point $\mathbf{q}_k, p_k \in \mathcal{N}_i$ (marked in red) for the same example as in Fig. 4.9 using a Gaussian RBF with different support parameters (growing support from left to right). The weight functions take the value 1 at the warp position itself and 0 at all other data points. However, non-negativity is not guaranteed.

to one:

$$w_n(\mathbf{q}_a) = \frac{f_n(\mathbf{q}_a)}{\sum_{p_t \in \mathcal{N}_i} f_t(\mathbf{q}_a)} \ . \tag{4.15}$$

If the new pose p_a is equal to an example pose p_n, this example gets an infinite weight f_n, and after normalization it is the sole contributor to the warp interpolation.

An example for kNN-interpolation of a weight function in a 2D pose-space is illustrated in Fig. 4.9. The images depict the weight function $w_k(\mathbf{q})$ of one example warp $\mathcal{W}_{i\to k}, p_k \in \mathcal{N}_i$. This warp is positioned at $\mathbf{q}_k$ in pose-space (marked in red). $\mathbf{q}_i$ is the data point in the center. The weight function $w_k(\mathbf{q})$ is interpolated from known values at positions $\mathbf{q}_n, p_n \in \mathcal{N}_i$ with

$$\begin{aligned} w_k(\mathbf{q}_k) &= 1 \quad \text{at the position marked in red} \\ w_k(\mathbf{q}_{n\neq k}) &= 0 \quad \text{at the positions marked in gray.} \end{aligned}$$

Fig. 4.10 illustrates interpolated weight functions for the same example achieved with Gaussian RBF interpolation. In this example, one positive characteristic of kNN-interpolation in contrast to RBF interpolation becomes visible: RBF interpolation can yield negative weight functions, which can result in exaggerated and unreliable warps, whereas kNN always guarantees non-negative weight functions, allowing for a more natural interpolation of warps.

Figure 4.11.: Blending before and after silhouette clean-up.

4.2.2. Image Blending

After the previous section has focused on the interpolation of warps to an arbitrary pose p_a for each of the selected example images $\mathcal{I}_i$, this section describes the final blending of the warped images to one result. The intensities are interpolated akin to the warp parameters by a linear blend of the warped images:

$$\mathcal{I}_a(\mathbf{x}) = \sum_{i=1}^{m} b_i(\mathbf{q}_a) \cdot \tilde{\mathcal{W}}_{i \to a}\big(\mathcal{I}_i(\mathbf{x})\big) \ , \tag{4.16}$$

where $b_i(\mathbf{q}_a)$ is a blending weight function for image $\mathcal{I}_i$ at the new pose position $\mathbf{q}_a$. This blending weight function is calculated with the k-nearest neighbors interpolation method with $k = m$ as detailed in the previous section.

For each of the selected images, the warp interpolation scheme results in a smooth animation sequence, when traveling through the pose-graph. However, the warps are interpolated from different sets of example poses for each of the images selected for blending. For each image, this set of examples is defined by its neighbors in the pose-graph. Hence, the final warped images to be blended might not be fully aligned. This is a consequence of different references and example positions for each of the interpolation problems. To assure photo-consistency during blending, the silhouettes of the warped images are adjusted to the silhouette of the image with the highest blending weight. The registration scheme of the silhouettes is simple and fast: the silhouettes are sampled, and a non-rigid iterative closest point (ICP) algorithm registers the silhouette point sets, similar to the silhouette ICP described in Sec. 4.1 on page 64. As the images are already close to each other, the ICP converges in fewer than 4 iterations in most cases. Fig. 4.11 compares a blending result before and after silhouette clean-up.

4.2.3. Definition of Subspaces

With the method described in the previous sections, plausible warps that can be interpolated from the example warps in the database are restricted to the convex hull of example warps, and the variety of possible pose images depends on the number of example images and warps per example as well as their position in pose-space. Because of the high dimensionality of the pose-space, it is therefore difficult to synthesize arbitrary pose images with a limited set of examples. Under the assumption that wrinkling is mostly affected by the nearest joints (e.g. the left elbow does not influence deformations at the right arm), the pose-space can be split up into subspaces, related to different parts of the body. These subspaces can have different pose-graphs, and each database image is parameterized in each subspace by the joint parameters belonging to the corresponding part of the body. In each subspace, the synthesis procedure is akin to the procedure for the full pose-space described in the previous sections: the closest m images to p_a are selected, and for each image, a warp to the new (full) pose is interpolated for each of these images before the warped images are blended. In contrast to the procedure described in the previous sections for the full body model, each vertex is now influenced by different subspaces, and warp parameters as well as intensities are interpolated locally with different weights for each vertex, as detailed below. The separation of the pose-space into subspaces reduces the dimensionality of the pose-space and allows for a larger variety of possible poses, without the need of more example images. Similar assumptions of local influences of joints also appear in [ACP02, RLN06, WHRO10].

To split up the pose-space into subspaces related to independent parts of the body, influence fields are defined, indicating how much a vertex is influenced by a specific part. In the presented experiments, the upper body (torso, sternum, shoulder and elbow joints), as well as the lower body (torso, hip and knee joints), are split up into two parts each, related to the left and right side of the body. Other partitions are possible, e.g. one subspace per limb. However, the number of images necessary for the synthesis of the final result increases with the number of subspaces. Hence, the number of subspaces is generally a trade-off between a larger variety of possible poses with fewer examples and the number of images to be warped and blended.

The influence fields of the different parts of the body should be smooth and overlapping. For the proposed partition, one possible approach to calculate the influence fields is based on the minimum distance to all joints of the respective body part. Let $d^L(\mathbf{v}_k)$ and $d^R(\mathbf{v}_k)$ denote the minimum distances of a vertex $\mathbf{v}_k$ to the left and right joints, respectively. Vertices with $d^L(\mathbf{v}_k) > d^R(\mathbf{v}_k) + \kappa$ are clearly assigned to the right part of the body and vertices with $d^R(\mathbf{v}_k) > d^L(\mathbf{v}_k) + \kappa$ to the left part of the body. To calculate the influence weights for the left part, the following linear system is solved (compare App. A.1.3 on Laplacian interpolation/approximation on a mesh):

$$\begin{bmatrix} \mathbf{D} \\ \lambda \mathbf{L} \end{bmatrix} \boldsymbol{\omega}^L = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} , \tag{4.17}$$

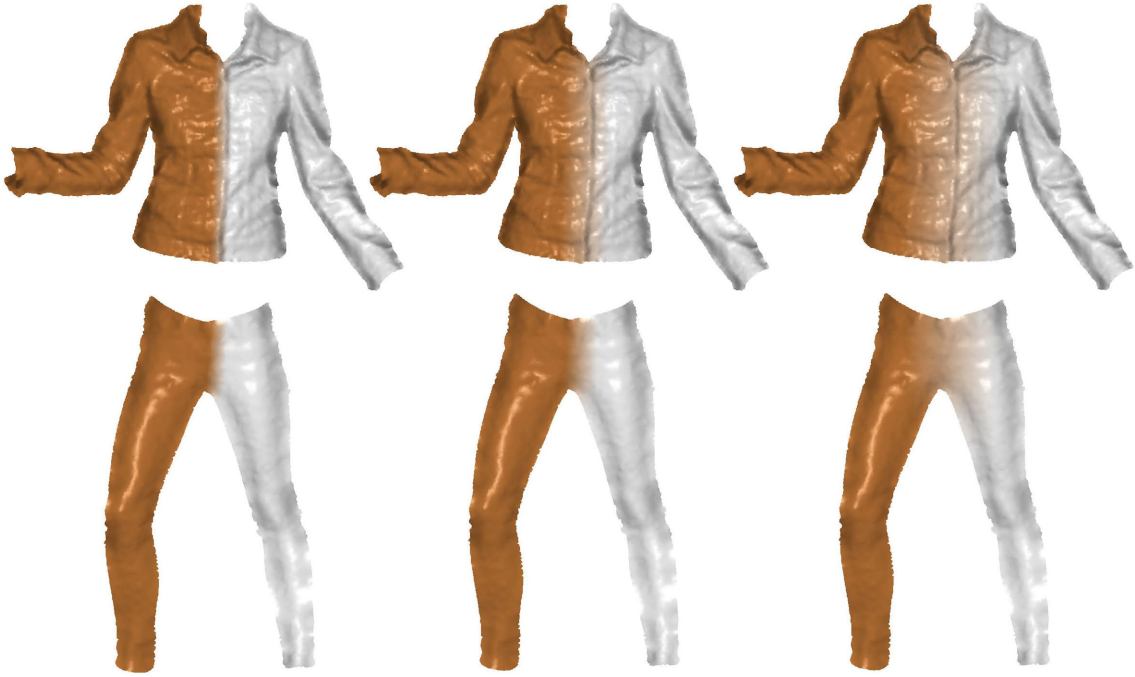


Figure 4.12.: Influence fields for the right upper and lower parts of the body on the mesh vertices for growing values of κ from left to right. Saturated colors indicate high values near 1, unsaturated colors indicate low values near 0.

with $\mathbf{D} = [\mathbf{I}\ \mathbf{0}]$, where $\mathbf{I}$ denotes an identity matrix of the size of the number of vertices clearly assigned to one side, $\mathbf{d}$ has entries 1 for vertices clearly assigned to the left part and 0 for vertices clearly assigned to the right part, and $\boldsymbol{\omega}^L = \begin{bmatrix}\omega_1^L & \dots & \omega_K^L\end{bmatrix}^T$ denotes the vector of vertex influence weights for the left part of the body. The weights and vertices are ordered accordingly, and $\mathbf{L}$ denotes a mesh Laplacian for smoothing over the complete mesh. The influence weights for the right part of the body can then be determined by $\boldsymbol{\omega}^R = \mathbf{1} - \boldsymbol{\omega}^L$. Both the parameter κ and the regularization parameter λ control on the *sharpness* of the influence fields. A small value for κ means a small region of *unknown* weights between the two parts, and the size of λ models the smoothness of the influence fields, i.e. the transition between the different parts of the body. Fig. 4.12 plots example influence fields of the right part of the upper and lower body for different values for κ.

The influence fields are used for local warp as well as intensity interpolation. For all selected database images, regardless in which subspace they were selected, a warp to the new (full) pose p_a is interpolated as explained in Sec. 4.2.1, where Equ. (4.10) is altered to

$$\begin{aligned}\Delta\tilde{\mathbf{v}}_{k,i\to a} &= \sum_s \sum_{p_n \in \mathcal{N}_i^s} w_n^s(\mathbf{q}_a) \cdot \omega_k^s \cdot \Delta\mathbf{v}_{k,i\to n} \\ \tilde{\rho}_{k,i\to a} &= \sum_s \sum_{p_n \in \mathcal{N}_i^s} w_n^s(\mathbf{q}_a) \cdot \omega_k^s \cdot \rho_{k,i\to n}\ .\end{aligned} \tag{4.18}$$

$\sum_s$ describes the summation over the different subspaces $s \in \{L, R\}$, and ω_k^s is

Figure 4.13.: A synthetic pose image (center) generated from two subspaces of the pose-space. The database images selected for the synthesis of the different body parts are depicted on the left (blending weights $b_1^1 = 0.35$ and $b_2^1 = 0.65$) and the right (blending weights $b_1^2 = 0.88$ and $b_2^2 = 0.12$).

Figure 4.14.: Synthesis of a completely new pose with pose-space partitioning (left, the two nearest database images for the left and right body parts are depicted on the left and right, respectively) and without local subspaces (right, the two nearest database images are depicted on the right).

the influence weight of subspace s on vertex $\mathbf{v}_k$. $\mathcal{N}_i^s$ denotes the neighborhood of pose p_i in the pose-graph of subspace s. After all images have been warped to the new pose, the images are blended locally based on the influence fields and blending weights per subspace. For this purpose, local blending maps $\mathcal{B}_{i\to a}^s(\mathbf{x})$ are interpolated by rendering the vertex influence weights at the warped vertex positions with barycentric interpolation between the vertices, and Equ. (4.16) is altered to

$$\mathcal{I}_a(\mathbf{x}) = \sum_s \sum_{i=1}^m b_i^s(\mathbf{q}_a) \cdot \mathcal{B}_{i\to a}^s(\mathbf{x}) \cdot \mathcal{W}_{i\to a}\big(\mathcal{I}_i(\mathbf{x})\big) \ , \tag{4.19}$$

where b_i^s denotes the blending weights for image $\mathcal{I}_i$ in the subspace s.

Fig. 4.13 shows a synthetic example pose image generated by local warping and

blending with $m = 3$. In the presented example, the right part of the body is composed of two images with blending weights $b_1^1 = 0.35$ and $b_2^1 = 0.65$ and the left part is composed of two different images with blending weights $b_1^2 = 0.88$ and $b_2^2 = 0.12$. More examples are shown in Sec. 4.3. Fig. 4.14 compares the synthesis of a completely new pose with and without pose-space partitioning. The left example shows a synthesis result with local subspaces for the left and right body parts. The two nearest database images for each body part are shown on the left and right, respectively. The right example shows the synthesis result without pose-space partitioning. Because no similar pose was captured in the database, without separating the pose-space into subspaces, the new pose has to be synthesized from less appropriate database images with larger distances in pose-space and less appropriate example warps, leading to strong deformations and improper texture, e.g. the wrinkling at the right leg is not natural compared to the results achieved with subspace synthesis on the left.

4.2.4. Distance Measures in Pose-Space

This section analyses appropriate distance measures for the selection of the *best matching* database images given a new input pose p_a. For each of the matched images, a warp to the new pose is interpolated from the stored database warps as detailed in Sec. 4.2.1 and 4.2.3, and the warped images are blended as explained in Sec. 4.2.2 and 4.2.3.

The simplest method to search for appropriate database images given a new pose configuration would be to measure the distances in pose-space and select those database poses, which minimize e.g. the L^2-norm of the pose vectors (Equ. (4.1)):

$$d_{ai}^{L^2} = \|\mathbf{q}_a - \mathbf{q}_i\| \ .$$

This metric reflects how close the poses are to each other in pose-space and, under the assumption that appearance is pose-dependent, how similar the images are to the appearance for the new pose. However, whether a suitable warp can be interpolated from a selected database image $\mathcal{I}_i$ does not only depend on its proximity to the new pose in pose-space but also on the number and positions of its neighbors, i.e. whether suitable warps are stored in the database for that image. Therefore, the stored database warps have to be taken into account.

With the interpolation scheme introduced in Sec. 4.2.1, a pose p_a, for which a warp can be interpolated from a sample pose p_i, is constrained by (see App. A.1.4)

$$\mathbf{q}_a = \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_n \ , \tag{4.20}$$

and plausible warps can be achieved for weights

$$\begin{aligned} 0 \leq & w_n \leq 1 \\ \sum_{p_n \in \mathcal{N}_i} & w_n = 1 \ . \end{aligned} \tag{4.21}$$

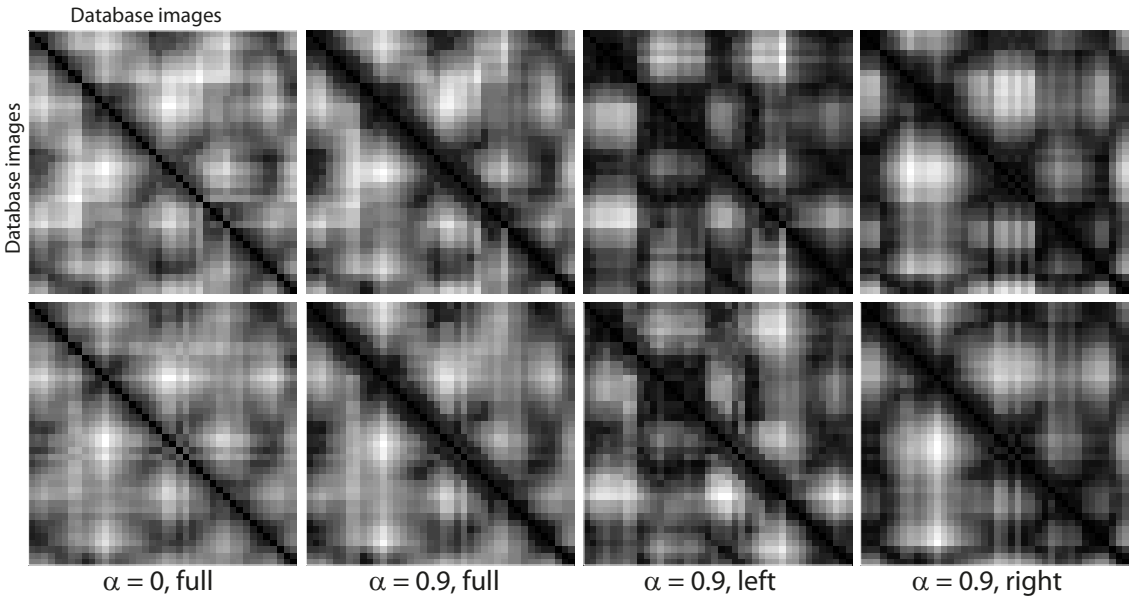


Figure 4.15.: Distance matrices between database images for $\alpha = 0$ for the full body model, as well as $\alpha = 0.9$ for the full, left and right body model. Dark colors indicate low values, bright colors indicate high values. Top: pose parameterization based on joint angles. Bottom: pose parameterization based on joint locations.

Hence, a distance measure taking into account the stored database warps can be defined as

$$d_{ai}^{\text{Warps}} = \|\mathbf{q}_a - \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_n\| \ , 0 \leq w_n \leq 1 \ . \tag{4.22}$$

Such a measure reflects how well a warp to the new position in pose-space can be interpolated from the stored warps. As the distance between the selected images and the new position in pose-space should not be discarded completely, the following combined distance measure is minimized:

$$d_{ai} = (1 - \alpha) \cdot d_{ai}^{L^2} + \alpha \cdot d_{ai}^{\text{Warps}} \ , \tag{4.23}$$

where $d_{ai}^{L^2}$ is a distance measure between the poses p_i and p_a according to Equ. (4.1), and α weights the influence of both terms. In general, $\alpha = 0$ favors images that lie close to the new position in pose-space. These images are similar to the new pose in terms of pose-dependent appearance, i.e. texture and shading. $\alpha = 1$ favors images with adequate example warps, regardless of their distance to the new pose in pose-space. In practice, example warps have been estimated between similar database images during the training phase, i.e. images close to each other in pose-space. Hence, images with adequate example warps will also lie close to the new pose and thus also exhibit adequate texture and shading. Hence, values close to 1 are used for α in all presented experiments ($\alpha = 0.9$). For illustration of the influence of α on the similarity measure, Fig. 4.15 plots the distances between the images of an example database for $\alpha = 0$ for the full body model as well as for $\alpha = 0.9$ for the

full, left and right body models. The upper row depicts the distances between the database images for pose parameterization based on joint angles, and the lower row shows the distances between the images based on joint locations. The distances are scaled in each matrix, brighter colors representing larger distances, and darker colors representing smaller distances, i.e. higher similarity. Compared to the distance field accounting only for the image positions in pose-space ($\alpha = 0$), the distance field is smoothed by additionally accounting for the database warps ($\alpha = 0.9$). Note the difference in the distance fields for the left and right body parts, compared to the full body part.

If the images for an animation sequences are selected independently from the database and no similarity between consecutively chosen database images is taken into account, sudden changes can appear in the set of selected database images, especially if the pose-space is sparsely populated with examples near the input poses. This can lead to jumping artifacts in the resulting synthesized animation sequence. These artifacts can be perceptually more disturbing than smooth transitions but with larger deformations. Therefore, a term reflecting the consistency between consecutively matched images may be included additionally:

$$d_{ai}^{\text{temp}} = (1 - \beta) \cdot d_{ai} + \beta \cdot d_{\text{prv}i}^{L^2} \ , \tag{4.24}$$

where $d_{\text{prv}i}^{L^2}$ measures the distance between database poses p_i and the previously selected pose with the highest blending weight p_{prv}. This distance can be precalculated and stored in the database as a matrix of inter-distances between the database poses (Fig. 4.15). An analysis of the different weights in the similarity measure is given in Sec. 4.3.

4.2.5. View Interpolation

Interpolation between views is performed akin to pose interpolation using kNN-interpolation. The distance is measured based on the viewing angle of the cameras in the torso coordinate system, denoted by α in the following (in other words, the pose-space for view interpolation is one-dimensional). Given a new pose and viewpoint, the nearest m_p poses are determined as described before, and the nearest m_v viewpoints are determined based on the viewing angles. For each of these viewpoints, the m_p pose images are selected, such that in total, $m_p \cdot m_v$ images are used for the synthesis. Let $\mathcal{I}_{i,j}$ denote a selected image for pose p_i and viewpoint v_j, and $b_i(\mathbf{q}_a)$ denote the blending weight of pose p_i for the new pose p_a and $b_j(\alpha_b)$ denote the blending weight of viewpoint v_j for the new viewpoint v_b, both calculated using kNN-interpolation. The blending weight for the image $\mathcal{I}_{i,j}$ is then given by:

$$b_{i,j}(\mathbf{q}_a, \alpha_a) = \frac{b_i(\mathbf{q}_a) \cdot b_j(\alpha_b)}{\sum_{s}^{m_p} \sum_{t}^{m_v} b_s(\mathbf{q}_a) \cdot b_t(\alpha_b)} \ . \tag{4.25}$$

For each image, the warp parameters are interpolated separately for pose and view changes. The final warp parameters are given by the sum of both interpolation results:

$$\begin{aligned} \Delta\tilde{\mathbf{V}}^{\mathcal{D}}_{i\to a,j\to b} &= \Delta\tilde{\mathbf{V}}^{\mathcal{D}}_{i\to a,j} + \Delta\tilde{\mathbf{V}}^{\mathcal{D}}_{i,j\to b} \\ \tilde{\boldsymbol{\rho}}^{\mathcal{D}}_{i\to a,j\to b} &= \tilde{\boldsymbol{\rho}}^{\mathcal{D}}_{i\to a,j} + \tilde{\boldsymbol{\rho}}^{\mathcal{D}}_{i,j\to b} \ . \end{aligned} \tag{4.26}$$

The separate interpolation has the advantage that only warps between different poses but the same viewpoint or warps between different viewpoints but the same pose need to be stored. Another approach would be to treat different viewpoints of the same pose as a new pose, i.e. each rigidly moved by the camera rotation and translation, and to include the orientation of the torso coordinate system into the pose representation. The disadvantage of this approach is that it would increase the pose-space by additional dimensions and that more warps need to be stored in the database, i.e. warps across viewpoints and poses.

4.3. Experiments and Results

A number of databases of upper and lower body clothing with different numbers of example poses and viewpoints has been created. An overview is given in Tab. A.3 on page 118. The number of database images varies from 33 to 320 (poses and viewpoints). In all databases and following experiments, two subspaces of the pose-space have been used, related to the left and right body parts as explained in Sec. 4.2.3. Pose-graphs have been established for each subspace separately based on the distance between the database images in the respective subspace. The average number of known warps between the database images varies from 6 (3 warps to different poses in the same viewpoint, 2 warps to the same pose but different viewpoints) to 33 (11 warps to different poses in the same viewpoint, 3 warps to the same pose but different viewpoints) per subspace, depending on the average distance in pose-space, i.e. the sparsity of the dataset.

Pose Synthesis and Animation. The presented approach has been tested on various animation sequences of upper and lower body clothing. Results are best evaluated visually. Fig. 4.16 shows example frames of animation sequences. Details of these frames are depicted in Fig. 4.17 on page 80. In these examples, the warps have been interpolated in pose-space using kNN-interpolation with $k = \min(10, |\mathcal{N}_i|)$. The number of images used for the synthesis is a trade-off between the number of pose-space interpolations to be performed and blurring artifacts introduced by more images on the one hand and temporal smoothness of the animation on the other. In the presented examples, $m_p = 5$ and $m_v = 3$ have been used. The use of real images and warp interpolation yields realistic movements of fine pose-dependent details during an animation, such as wrinkles (e.g. wrinkling at the elbows in the jacket sequence or at the knee and the inner regions of the legs in the jeans sequence), fine details (such as the epaulette in the shirt sequence) as well as shading (e.g. shadow

Figure 4.16.: Example frames of synthetic animation sequences. Fine details (see also Fig. 4.17), such as wrinkling at the elbow, shading and the complex movements, e.g. of an epaulette, can be modeled with the proposed approach.

Figure 4.17.: Details of the synthetic animation frames shown in Fig. 4.16.

Figure 4.18.: Examples of pose extrapolation. To illustrate the degree of extrapolation, in each example, the left-most images depict the nearest interpolated pose, and the center and right images show extrapolated synthetic results. Details of each example are shown in Fig. 4.19.

cast by the arm on the body in the jacket sequence). Even for loose clothing (such as the blouse), strong wrinkling at the torso induced by arm movements can be modeled.

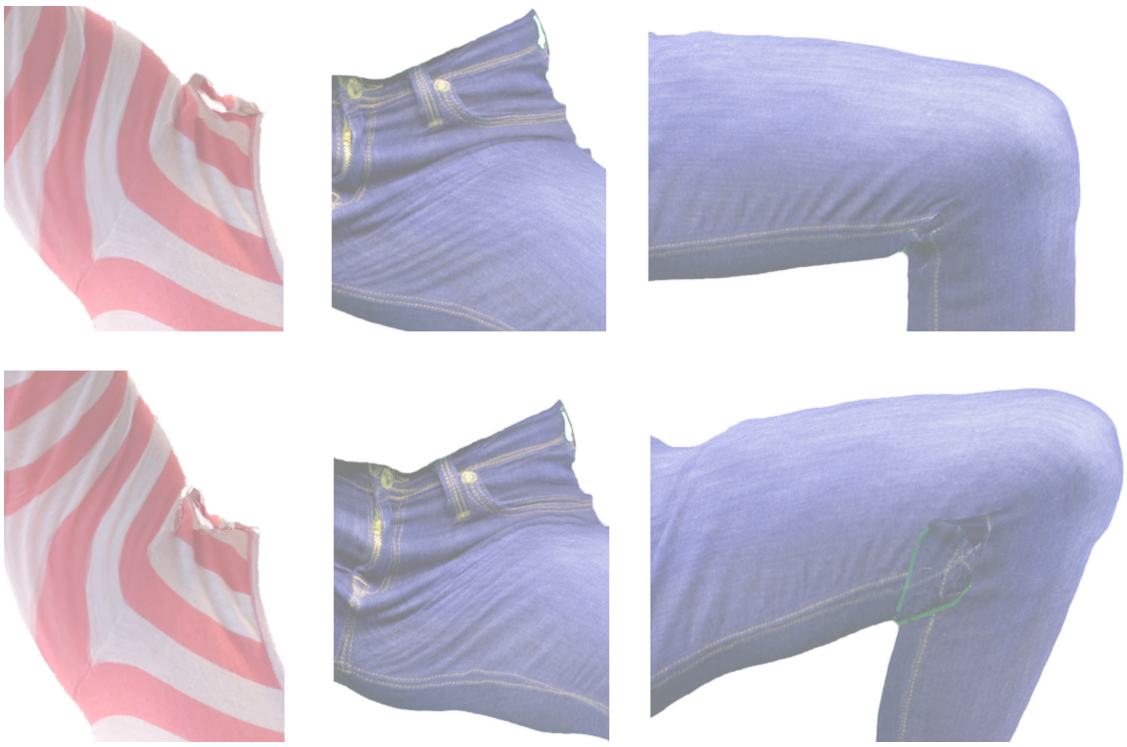


Figure 4.19.: Details of the extrapolation examples in Fig. 4.18.

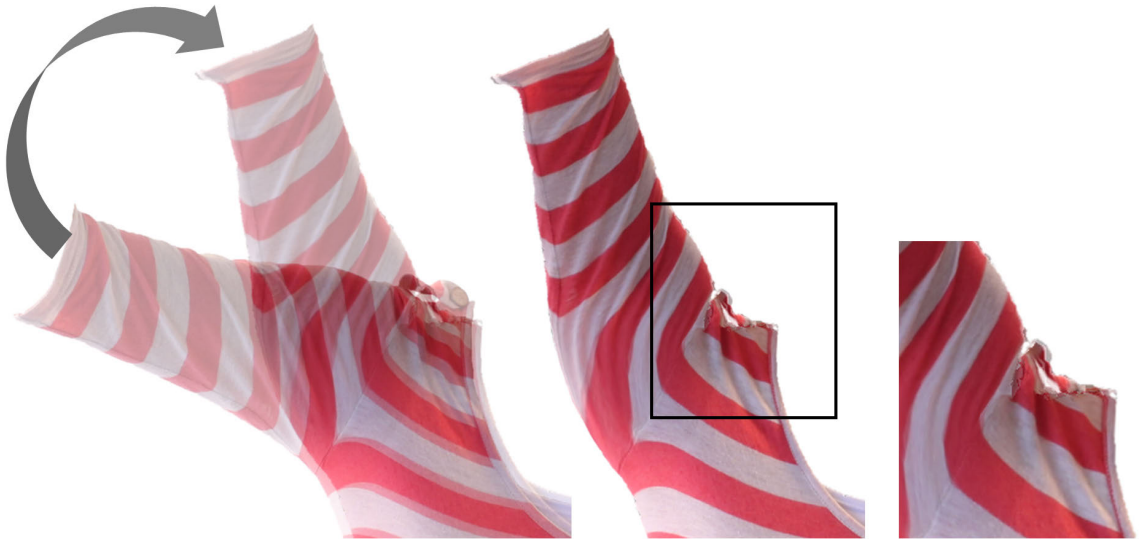


Figure 4.20.: Collapsing joint artifact induced by skeletal subspace deformation with linear blend skinning for strong extrapolation.

Fig. 4.13 and 4.14 on page 74 show examples of completely new synthetic poses, in which the left and right body parts have been synthesized in different subspaces of the pose-space from different database images. The first two matched database images used for the synthesis in each subspace are depicted on the respective side of the synthetic results. No similar pose has been captured in the database for the full body, and without splitting up the pose-space into subspaces, more example poses would have been necessary to realistically synthesize these poses. This is demonstrated in Fig. 4.14, which directly compares the synthesis result using two subspaces to a synthesis result using one pose-space for the full body model. Because no similar pose is present in the database, without pose-space partitioning, the new pose has to be synthesized from less appropriate database images, leading to unnatural deformation and improper texture.

Pose Extrapolation. Besides interpolation, experiments on pose extrapolation have been conducted. Example frames of extrapolated animations are depicted in Fig. 4.18 on page 81. To illustrate the degree of extrapolation, in each example, the left-most images depict the nearest interpolated pose, and the center and right images show extrapolated poses. Details of these pose extrapolation examples are shown in Fig. 4.19. These examples show that the kNN-warp-interpolation scheme yields plausible results for moderate extrapolation.

One reason for this is that kNN-interpolation always produces non-negative weights such that warps are not exaggerated. Hence, wrinkling and shading can even be synthesized realistically for extrapolation, as long as deformations from SSD-warping can be interpolated linearly. For example, the wrinkles at the upper legs of the jeans or the epaulette of the shirt move realistically and naturally in the extrapolated sequence. However, for stronger extrapolation, the assumption that deformations from SSD-warping can be inter- or extrapolated linearly is not valid anymore. The reason for this is that SSD with linear blend skinning (LBS) models deformation as a linear combination of rigid transformations, which is only a valid assumption for small pose changes. For extreme pose changes, i.e. larger joint rotations, SSD with LBS produces unnatural deformation and tends to introduce artifacts, e.g. the collapsing joint artifact, where the deformed mesh loses volume as the joint rotation increases [LCF00] (see Fig. 4.20). If these deformations are too large, they cannot be removed by a linear combination of stored detail-warps anymore. Hence, the larger the distance between the extrapolated pose and the example poses, the more unnatural deformations are introduced, which cannot be corrected by the database warps. Also, with stronger extrapolation, blurring artifacts can appear due to misalignment of the images as no warp examples are known outside the convex hull (Fig. 4.19). Finally, self-occlusions in the extrapolated pose that have not been captured in the database can cause overlapping artifacts during an extrapolated animation (see knee detail in Fig. 4.19).

Similarity Measure. Fig. 4.21 illustrates the influence of the similarity measures introduced in Sec. 4.2.4 on the visual quality of the synthesized results. It plots three frames of an animation sequence. In the top row, the database images used for the synthesis have been selected based on pose similarity only, and no warp directions are

Figure 4.21.: Influence of a warp consideration term in the similarity measure on three synthesized frames in an animation sequence. Top row: similarity measure solely based on pose distance ($\alpha = 0$ in Equ. (4.23)). Bottom row: similarity measure additionally accounting for the database warps ($\alpha = 0.9$ in Equ. (4.23)).

taken into account ($\alpha = 0$ in Equ. (4.23)). The bottom row shows results achieved with additional warp direction consideration in the similarity measure ($\alpha = 0.9$ in Equ. (4.23)). If the database warps stored for each database image are not taken into account during matching, database images with improper warps can lead to a poor animation result with unnatural deformations (top row). The visual quality of the animation is much higher when database warps are taken into account in the similarity measure (bottom row).

The influence of the temporal consistency term in the similarity measure ($\beta > 0$ in Equ. (4.24)) can be seen from Fig. 4.22-4.23. The figures plot the measured distances between poses of a new animation sequence (horizontal axis) to the database

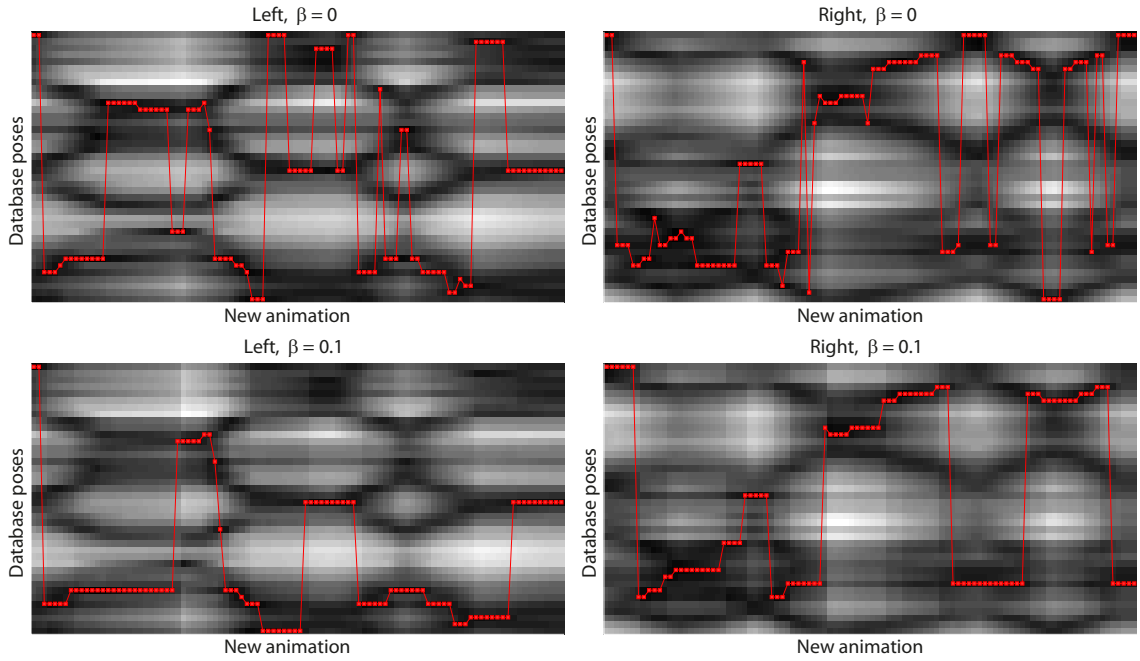


Figure 4.22.: Influence of a temporal consistency term on the selection of database images during an animation sequence for a parameterization based on joint angles. A temporal consistency term ($\beta > 0$ in Equ. (4.24)) helps to prevent sudden jumps and oscillations in the sequence of matched database images.

images (vertical axis) of the same database as depicted in Fig. 4.15 on page 76. The distances are color coded, darker colors representing small pose distances (i.e. higher similarity), brighter colors representing larger pose distances. The top row depicts similarity matrices for $\beta = 0$, and the bottom row shows similarity matrices for $\beta = 0.1$ ($\alpha = 0.9$ in all examples), both for the two subspaces, related to the left (left column) and the right (right column) body parts. The graphs plot the minimum for each pose in the new animation sequence, i.e. the matched poses with the highest blending weight used for the synthesis. Fig. 4.22-4.23 show the results for the same animation sequence with pose parameterization based on joint angles (Fig. 4.22) and joint locations (Fig. 4.23). Depending on the parameterization, a slightly different sequence of database images is matched. In the presented example, sudden jumps and oscillations can be prevented with a weight $\beta = 0.1$ to the temporal consistency term. However, this temporal smoothness can come at the cost of inferior matches to the input pose, requiring more deformation. For this reason, the weight β has to be chosen carefully and should be small ($\beta = 0.1$ in the presented experiments).

Database Size and Sampling Density. The question of how many poses are needed for a *complete* or *sufficient* database is not a trivial one. In general, the interpolation domain is restricted by the examples in the database (per subspace/body part), which means that at least *similar* poses to the poses to be synthesized must

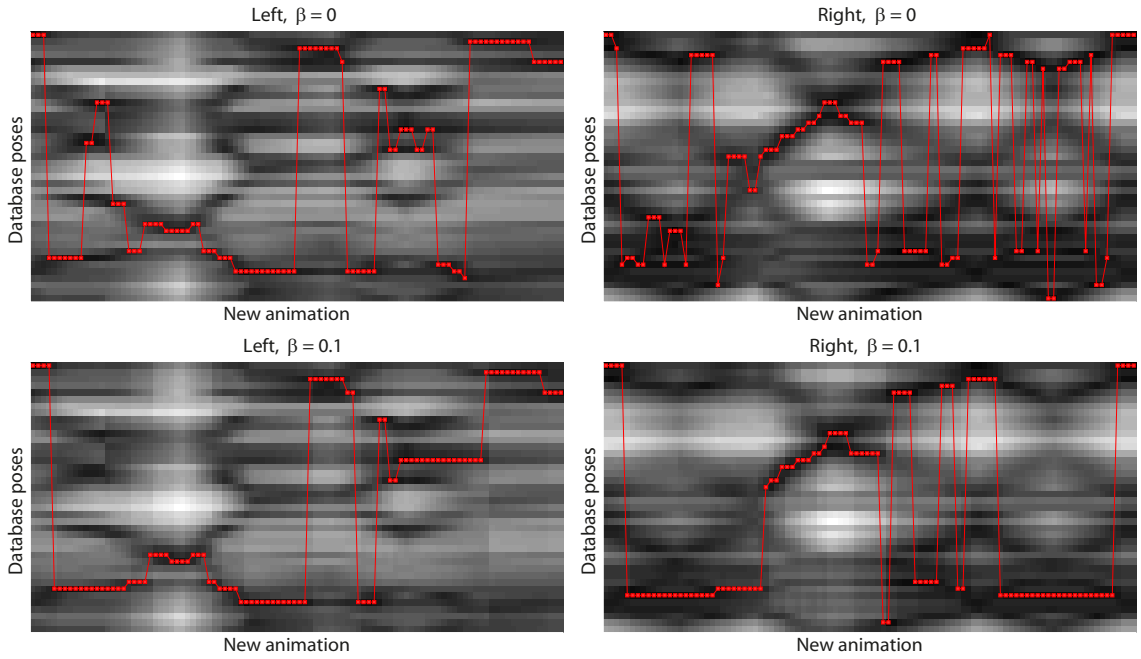


Figure 4.23.: Influence of a temporal consistency term on the matching of database images for the same animation sequence as in Fig. 4.22 for a parameterization based on joint locations. Compared to a pose representation based on joint angles, a slightly different sequence of database images is matched.

be available in the database (again per body part/subspace). A dense sampling facilitates both warp estimation in the preprocessing phase as well as warp interpolation in the rendering phase. As already addressed above, for large pose animations, SSD with LBS might introduce unnatural deformations and artifacts, such as the collapsing joint artifact [LCF00]. This happens because LBS assumes that deformations can be modeled as a linear combination of joint transformations, which is a valid assumption only for small joint rotations, i.e. small pose changes. Hence, the larger the distances between the database images in pose-space, the larger are the deformations and shading differences from SSD-warping that must be corrected by the detail-warps, making image registration and warp estimation more challenging. Furthermore, if the poses in the database differ too much, also *contextual* differences between the images increase, e.g. differences in texture, wrinkling and shading, and the less accurate can one image be registered onto another. Self-occlusions in the database images, for example, can lead to missing texture information such that the texture in this region is stretched when deformed onto another pose (compare e.g. Fig. 4.24). Finally, the a dense sampling of the pose-space by examples facilitates smooth transitions between poses in an animation sequence. Generally, the required sampling density of the pose-space depends on the complexity of poses and the piece of captured clothing. Looser clothing will need a much more dense sampling than very tight fitting clothing.

Figure 4.24.: The larger the distances between pose images in the database the more challenging is the accurate registration with image warps. Especially, in case of self-occlusions in the source image, artifacts can appear in the warped database images. From left to right: source image; target image; warped source image onto target pose.

Fig. 4.25-4.26 on pages 88-89 illustrate the influence of the database size and the sampling density on the synthesis quality based on a leave-one-out experiment. One of the pose images (together with with pose information, geometry and warps from and to that pose, see Fig. 4.27 on page 90) was taken out of the database and compared with the synthetically generated image for that pose. The experiment was conducted with various sparse example databases, reducing the data by half for each experiment (Fig. 4.27). As shown in Fig. 4.25-4.26, the difference between the ground truth image and the synthetic image increases with a reduced database as (i) intensities and warps are interpolated from larger distances and (ii) during the analysis procedure the warps need to be estimated between more distant example images (compare Fig. 4.24). Nevertheless, the synthetic images, especially the pose-dependent characteristics, such as wrinkling and shading, are still visually correct as long as plausible warps can be estimated between the example images (recall that the objective is a realistic and plausible visualization of pose-dependent characteristics and not accurate reconstruction). With decreasing database size and density, the synthesis with pose-space partitioning is closer to the ground truth images than the synthesis without pose-space partitioning, as each body part can be synthesized from different body parts, leading to more appropriate texture and deformation.

Limitations. Like in any example-based or image-based method, the variety of poses that can be synthesized depends on the number of examples in the database and their distribution/density in pose-space. As discussed above, a dense sampling of the pose-space facilitates warp estimation as well as image synthesis, and the rendering quality depends the quality of the estimated image warps. Also, LBS-

Figure 4.25.: Comparison to ground truth (i). From left to right: synthetic pose images produced with the full, half and quarter set of example images and the according difference images between the ground truth and the synthetic images. Top: results without pose-space partitioning. Bottom: results with two subspaces.

SSD as the underlying animation technique might introduce artifacts for extreme pose changes, making correction by the detail-warps difficult. However, the proposed approach is not constrained to linear blend skinning but can be combined with any other more sophisticated animation method.

Image-based methods generally have the limitation that the object to be rendered has to be captured and processed in advance and no modification of appearance is possible afterwards, e.g. changes in texture or material, which can be easily exchanged in classical computer graphics rendering. Chapter 5 presents an image-based retexturing approach, which can be combined with the presented image-based rendering approach in future work to allow for additional appearance modification.

Cost. The presented approach shifts the computational complexity from the rendering phase to an a-priori training and data analysis phase. The cost for synthesis and visualization is low and mainly depends on the number of pose subspaces and

Figure 4.26.: Comparison to ground truth (ii). From left to right: synthetic pose images produced with the full, half and quarter set of example images and the according difference images between the ground truth and the synthetic images. Top: results without pose-space partitioning. Bottom: results with two subspaces.

selected images per subspace, which define the number of image warps and silhouette clean-ups to be performed. In contrast to radial basis function interpolation used in many PSD approaches, the k-nearest neighbors interpolation method is cheap, and the only time-consuming part is the warping of the images, which can be done in a few milliseconds. The size of the database influences the cost of the rendering phase only in the database image selection step, as the size of the database defines the search space. However, as temporal consistency between subsequent frames is desirable, for very large databases, the search space can be reduced to the nearest database images of previously selected frames, which can be pre-calculated and stored in the database.

The low costs during rendering come with high costs of data acquisition and storage requirements. Each individual object has to be captured in advance and for each provided view and pose, a 3D model and an RGB-alpha image need to be stored,

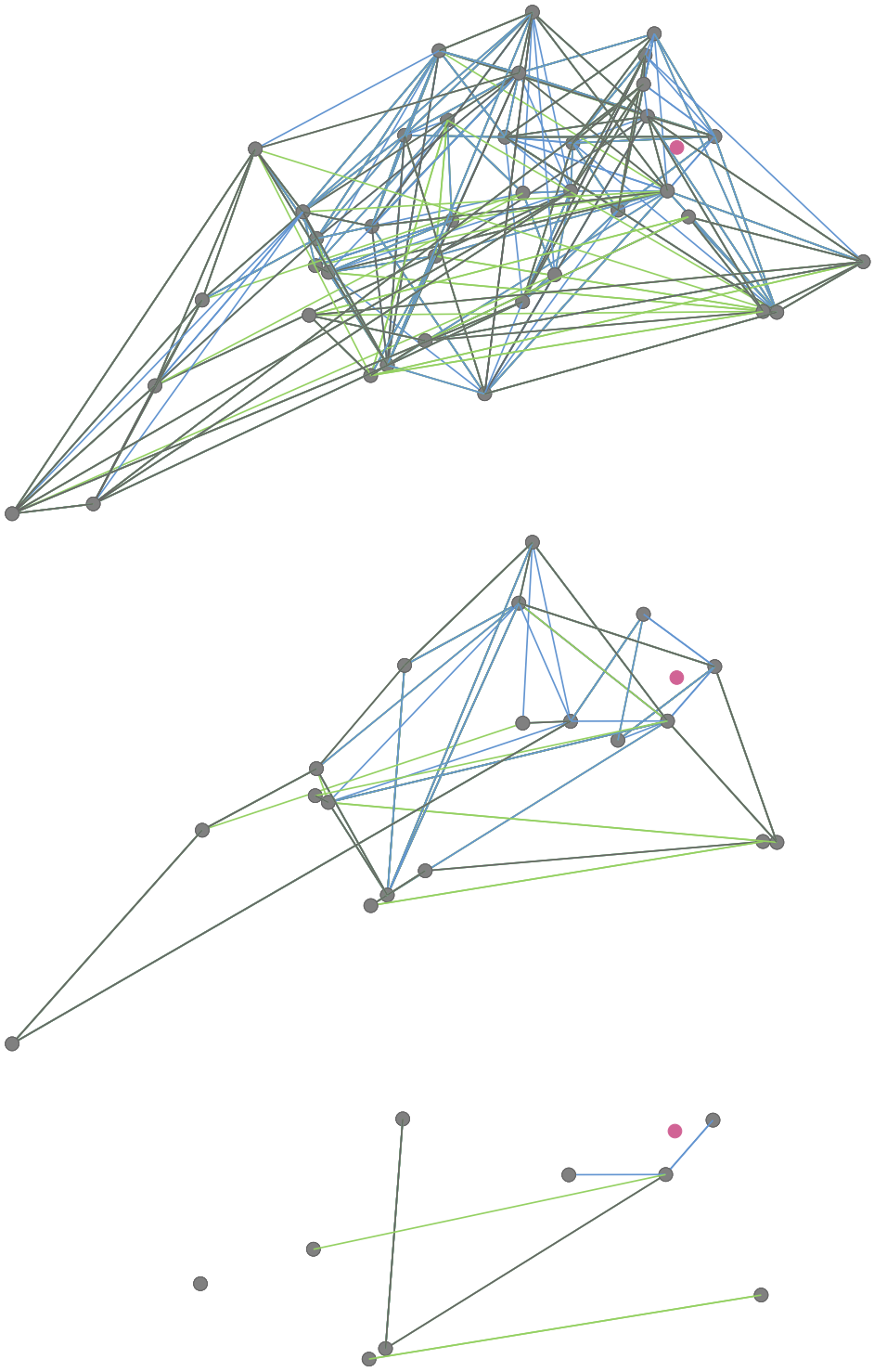


Figure 4.27.: Illustration of the the reduction of the database in the leave-one-out experiment (Fig. 4.25-4.26). From top to bottom: full, half and quarter databases. Note that not only poses but also all warp information to and from these poses are taken out of the database in the reduction. In the graphs, gray edges mark warps for both subspaces, blue edges mark warps only for the left body part, and green edges mark warps only for the red body part.

as well as sets of warp parameters and skinning weights. The high storage costs can be addressed by combining the proposed approach with an image-based retexturing method, such as the one presented in Chapter 5. Including absolute texture deformation and shading information into the database would allow for appearance changes in the database images, such that a piece of clothing needs not to be captured and stored for each individual design or cloth pattern.

4.4. Chapter Summary

This chapter has presented a new pose-dependent image-based rendering technique, which synthesizes new images of a piece of clothing based on the articulated pose of a human body by interpolating and merging clothing appearance from a database of images. In contrast to classical image-based rendering techniques, which are usually limited to viewpoint interpolation, complex animation sequences can be synthesized with the proposed method. This is achieved by interpolating image warps and intensities in pose-space using scattered data interpolation methods. The high dimensionality of the interpolation domain is addressed by partitioning the pose-space into subspaces related to body parts that can be handled independently, thereby reducing the dimensionality of the interpolation domain.

Experiments have shown that the concept of pose-dependent warp and intensity interpolation results in a realistic visualization of the clothes especially at fine pose-dependent details both for interpolation between poses as well as moderate extrapolations. Limitations and required sampling density of the pose-space have been discussed. Generally, the required sampling density of the pose-space depends on the complexity of poses and the piece of captured clothing. Experiments and analyses have been performed based on databases with a limited number of poses and demonstrate a *proof of concept*. With automatic segmentation and pose estimation methods, larger databases with a larger variety of poses can automatically be generated.

5. Image-Based Retexturing

The previous chapter has introduced an image-based rendering approach that allows modification and animation of an object, in contrast to classical image-based rendering that is restricted to rigid objects and viewpoint change. One remaining drawback of image-based methods in contrast to classical computer graphics rendering is that each individual object needs to be represented by a large number of images, and later modification of the texture and surface details is not possible. This chapter introduces an image-based method that allows appearance changes of an object in a given image by *retexturing* [HSE10, HSE11a, HSE11c]. In future work, the two approaches can be combined to additionally include texture modification information into the database representation of Chapter 4 for additional appearance modification.

Retexturing means the augmentation of a surface or object in an image with a new synthetic texture (Fig. 5.1). To assure that the virtual texture merges with the real image content, not only geometric properties such as position, pose and deformation of the original texture have to be preserved in the synthetic image but also photometric characteristics such as shading and lighting variations on the texture. The proposed approach works fully image-based and does not require any information about the 3D shape of the object. The virtual texture is blended into the original image such that its deformation as well as lighting conditions and shading are maintained from the input image. For this purpose, an input image $\mathcal{I}$ is modeled as a spatially and photometrically warped reference texture $\mathcal{T}$:

$$\mathcal{I}(\mathbf{x}) = \mathcal{W}_p\big(\mathcal{T}\big(\mathcal{W}_s(\mathbf{x})\big)\big) \ . \tag{5.1}$$

If the texture $\mathcal{T}$ is known, the warps can directly be estimated with the image-based warp optimization approach of Chapter 3. Once the warps have been extracted, they can be applied to any new texture, which is then blended into the original image via alpha-blending. Sec. 3.3.1 has presented an application of the warp optimization framework to tracking and retexturing of deforming surfaces in monocular video sequences, assuming that the first video frame provides the appearance of $\mathcal{T}$ and thus can be used as a reference texture.

A reference image of an undeformed texture $\mathcal{T}$, however, is not always available and for many applications difficult to provide, e.g. for a complete piece of clothing. Besides a general formulation of image-based retexturing, this chapter presents an approach to estimate texture deformation and shading as well as the appearance of the reference texture $\mathcal{T}$ from a single image under the assumption that $\mathcal{T}$ is of a regular type. A regular texture is constructed by regularly tiling the texture

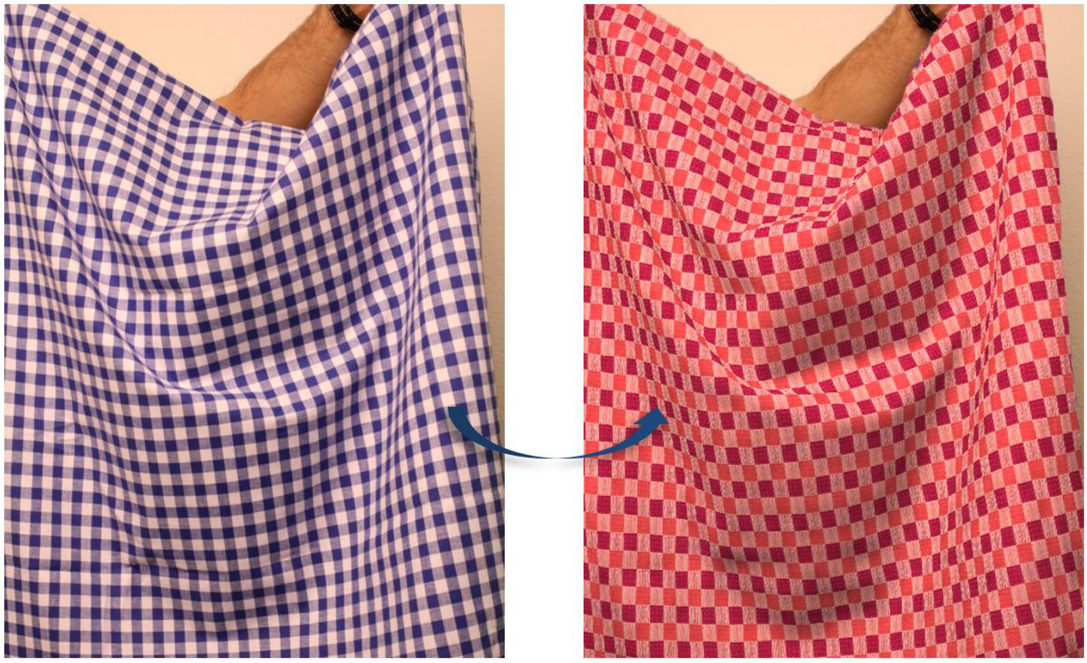


Figure 5.1.: *Retexturing* or *texture replacement* means the augmentation of an object in an image with a new synthetic texture such that texture deformation and shading properties are maintained from the original image.

space with the same texture element, called *texel* or *texton* [GS86] (Fig. 2.1 on page 20). In this case, $\mathcal{I}$ shows a texture that deviates from a regular congruent tiling both spatially and photometrically and is often called a *near-regular texture* (NRT) [LLH04, LL03]. The proposed approach exploits this assumption to decompose an image $\mathcal{I}$ of a near-regular texture into the appearance of the underlying regular texture as well as a deformation field and a shading map that transform the regular texture into the given near-regular texture. This decomposition is illustrated in Fig. 5.3 on page 96. The deformation field and the shading map are extracted as a joint spatial and photometric warp (Chapter 3) that registers a synthetic reference image of the regular texture $\mathcal{T}$ texture with the input image $\mathcal{I}$.

This chapter is structured as follows. Sec. 5.1 briefly introduces regular and near-regular textures. Sec. 5.2 presents a method for near-regular texture analysis and decomposition. This method allows retexturing of a single image without the need of a reference image. Sec. 5.3 presents results of the proposed NRT analysis and retexturing method.

5.1. Regular and Near-Regular Textures

Regular textures can be generated by tiling the texture space with one or more repeating texture elements. Mathematically, such patterns can be defined by one pattern element (called *texture element*, *texels* or *textons*) and two smallest linearly

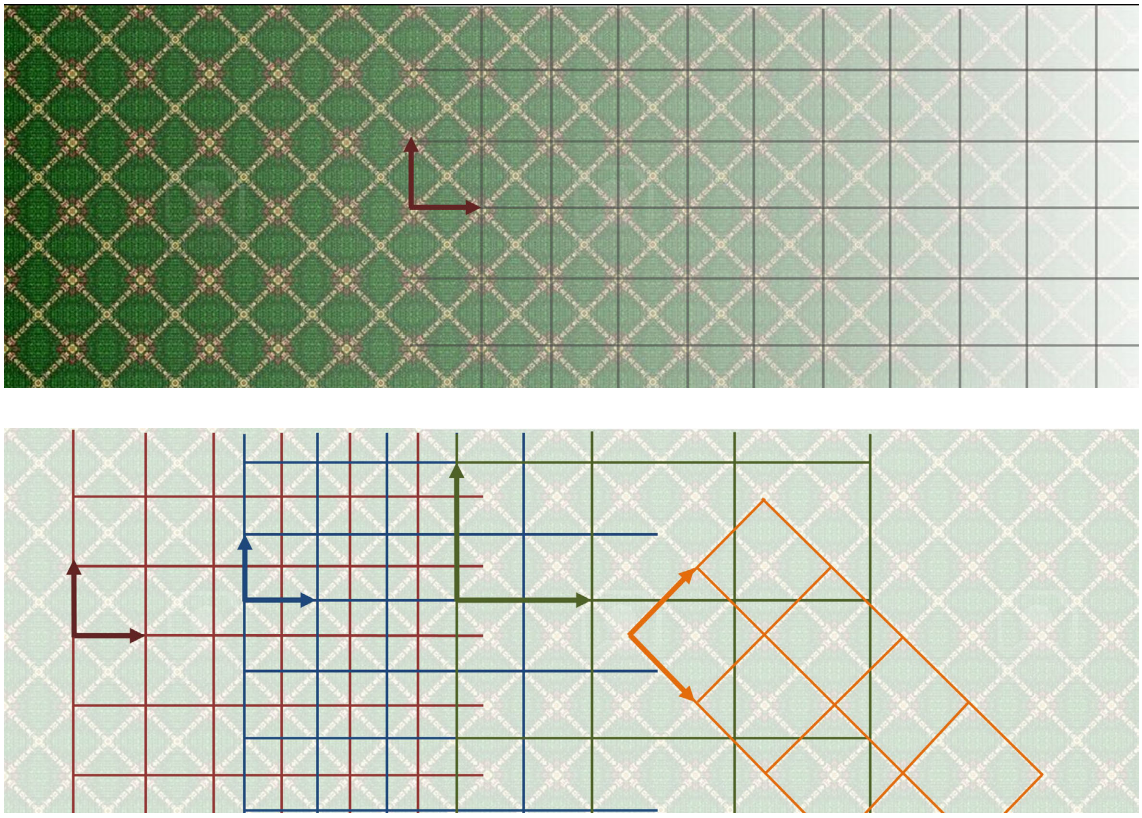


Figure 5.2.: Regular textures can be constructed by tiling the space with the same texture element using two generating vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ (top). In each regular texture, there are more than one valid texels and tiling patterns (bottom).

independent generating vectors $\mathbf{t}_1, \mathbf{t}_2$, describing the tiling pattern. This results in a degree-4 lattice, where each pattern element is a node with four neighbors representing its own copies at an offset of $\pm\mathbf{t}_1$ and $\pm\mathbf{t}_2$ [GS86, PBCL09] (Fig. 5.2). For each regular texture, there exists more than one valid texel and tiling pattern because each shifted version of a valid texel or a set of more than one valid texel also represent valid texels (Fig. 5.2).

Textures that deviate geometrically and/or photometrically from a regular congruent tiling are often called *near-regular textures* (NRT) [LLH04, LTL05] (see also Tab. 2.1 on page 20). In NRT, the texture elements appear geometrically and photometrically distorted, but they still exhibit the same topological regularities and relations as their regular counterpart [LLH04, PBCL09] (Fig. 5.1). Hence, they can be regarded as spatially and/or photometrically warped regular textures. For example, if a regularly textured 3D object is projected into an image, the texture appears as a near-regular texture in the image because of variations in the viewing angle and lighting conditions for each texel (Fig. 5.2). Based on these properties of regular and near-regular textures, the following section proposes a texture analysis approach for NRT, formulating the geometric and photometric deviations of near-regular textures as spatial and photometric image warps of a regular texture. The appearance of the

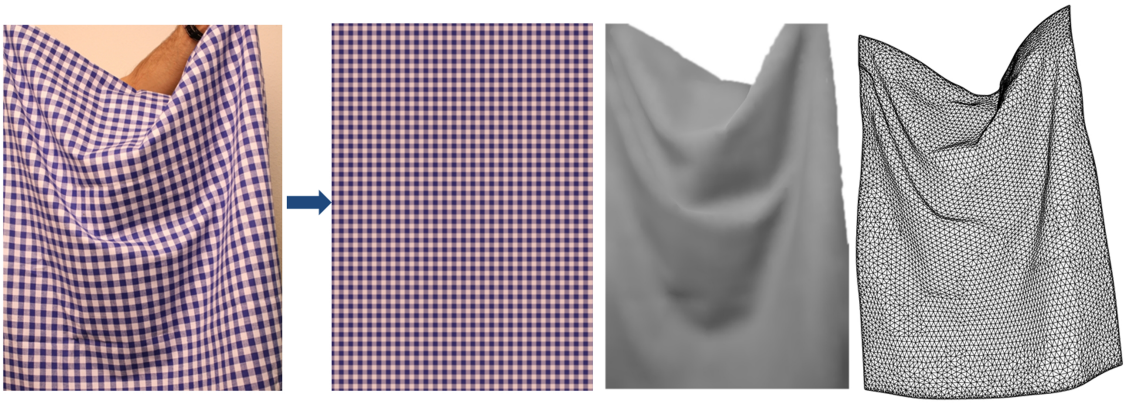


Figure 5.3.: Decomposition of the original image $\mathcal{I}$ (left) according to Equ. (5.2) into a regular texture $\mathcal{T}(\mathbf{x})$, a photometric image warp $\mathcal{W}_p(\mathcal{I})$, represented as a shading map, and a spatial image warp $\mathcal{W}_s(\mathbf{x})$, represented by a deformed mesh (see Chapter 3).

regular texture is estimated in a previous texture analysis step.

The proposed method requires two properties of the texture (similar to [LF06, PBCL09]):

- Regular repetition. A texture element must be repeated regularly (topologically) and often enough to provide a useful estimate of the texel appearance.
- Localizability. An element must have sufficient structure such that it can be easily localized in the image using common interest point detectors and transformations between texture elements can be estimated.

5.2. Near-Regular Texture Analysis and Decomposition

Following the previous analysis of regular and near-regular textures, an input image $\mathcal{I}$ of a near-regular texture is modeled as a spatially and photometrically warped regular texture $\mathcal{T}$

$$\mathcal{I}(\mathbf{x}) = \mathcal{W}_p\big(\mathcal{T}(\mathcal{W}_s(\mathbf{x}))\big) \ , \tag{5.2}$$

where $\mathcal{T}$ denotes the original regular texture, which is deformed by a spatial warp $\mathcal{W}_s(\mathbf{x})$, and $\mathcal{W}_p(\mathcal{T}(\mathbf{x}))$ denotes a photometric warp, e.g. a shading map multiplied to the texture intensities (see Chapter 3). This section describes, how the components $\mathcal{T}$, $\mathcal{W}_s$, $\mathcal{W}_p$ are estimated from the input image $\mathcal{I}$ under the assumption that $\mathcal{T}$ is regular (Fig. 5.3). Once the individual components have been estimated, $\mathcal{T}$ can be substituted by any new texture to create a synthetic image with a different texture but the same deformations and lighting properties). The proposed method consists of the following substeps:

- **Mean Texel Appearance and Lattice Estimation** (Sec. 5.2.1). Assuming that the undeformed texture $\mathcal{T}$ is regular, it can be constructed by regularly tiling the texture space with the same texel. Thus, the deformed texture $\mathcal{W}_p(\mathcal{T}(\mathcal{W}_s(\mathbf{x})))$ also exhibits repeated similar elements (up to spatial and photometric deformation) with the same topological relations. A first step estimates the mean appearance of a repeating texture element and a lattice structure, representing the topological relations between candidate texel positions in the image. From the mean texel appearance, an estimated appearance of the regular texture $\mathcal{T}$ of arbitrary size can be synthesized.
- **Warp-based Texture Decomposition** (Sec. 5.2.2). The spatial and photometric warps $\mathcal{W}_s$ and $\mathcal{W}_p$ between $\mathcal{T}$ and $\mathcal{I}$ are jointly optimized in an image-based warp optimization procedure using the warp optimization method presented in Chapter 3, providing information on texture deformation and shading.
- **Lattice Propagation and Fusion** (Sec. 5.2.3). After each optimization, the lattice is grown outwards and a new optimization round starts.
- **Texture Replacement** (Sec. 5.2.4). Having decomposed the image into its intrinsic parts, an arbitrary new texture (not necessarily a regular one) can now be substituted into Equ. (5.2) to produce an image of this texture with the same deformation and illumination properties as in the input image.

5.2.1. Mean Texel Appearance and Lattice Estimation

The first step in the NRT analysis and decomposition approach is to estimate the mean appearance of a repeating texture element as well as candidate positions of this texture element in the image, represented by a quadrilateral lattice. This texture element is used to synthesize the appearance of the underlying regular texture $\mathcal{T}$, serving as a reference image in the following warp optimization (Sec. 5.2.2). In contrast to regular textures, in NRT images, the texel appearances are no longer copies of each other but rather appear spatially deformed and of different colors. Nevertheless, the texel appearances are still *similar* (e.g. up to spatial and photometric distortion). In this section, a texel detection and lattice estimation approach exploiting these facts is proposed.

Detection of Repeating Image Structures. In this stage, low level image features are used to detect and cluster repeating elements in the image $\mathcal{I}$. For this purpose, suitable feature descriptors are generated on the image, e.g. using SIFT [Low03], and grouped using an unsupervised non-parametric feature-space analysis technique, e.g mean shift clustering [CM02] (Fig. 5.4 on page 99). Generally, any interest point detector and feature descriptor can be used. The choice of the descriptor is a trade-off between finding enough feature points to identify repeating structures in the image on the one hand and creating too many false positive detection results on the other. Although the texture in the image may be strongly distorted,

the idea is that transformations between local image patches can be approximated by a projective, an affine or even a similarity transformation. This assumption is similar to the planarity assumption used in most shape from texture methods [LH05, LF06, HSE11b]. Ideally, the feature descriptor should be invariant against small projective transformations. Although the SIFT descriptor is only invariant against rotation and scale and not against affine or perspective distortions, which appear at strong deformations, it produces good detection results if there are not too many strongly distorted texels. The feature descriptors are grouped together using mean shift clustering [CM02], which is an unsupervised non-parametric clustering approach, yielding clusters $\mathcal{C}_i : \{\mathbf{p}_j, j = 1 \dots N_i\}$ of image points $\mathbf{p}_j$ with similar SIFT descriptors, i.e. similar appearances (Fig. 5.4). The idea is that each cluster now contains instances of the same point on a repeating texel and that by finding a topological structure between these points, the underlying texture grid can be determined. The left image in Fig. 5.4 illustrates an example output of this step with different clusters marked in different colors.

The described texel detection strategy is similar to the texel identification of Park et al. [PBCL09]. However, in [PBCL09] normalized image patches of a fixed size are used as descriptors, which are only invariant against translation such that strongly rotated or scaled image patches cannot be detected.

Lattice Estimation. The interest points of each cluster should be related to each other by the same topological regularities and relations as in an undeformed regular texture. According to [LLH04, PBCL09], any deformed regular texture can be described by a degree-4-lattice, representing the tiling pattern of the texels in the undeformed texture. Thus, the interest points $\mathbf{p}_j$ of each cluster $\mathcal{C}_i$ are candidate vertices of a quadrilateral lattice, representing the tiling pattern (Fig. 5.4). The aim of this step is to estimate for each cluster a quadrilateral lattice model $\mathcal{L}_i : \{\mathbf{P}_i, \mathcal{Q}_i\}$, consisting of the cluster points $\mathbf{P}_i = [\mathbf{p}_1 \ \dots \ \mathbf{p}_N], \mathbf{p}_j \in \mathcal{C}_i$ and a quadrilateral topology $\mathcal{Q}_i$ between them, consistent with the spatial relationship between the interest points $\mathbf{p}_j$ and the assumed texture topology. In this lattice, the texture lying in each quad approximates a texel instance $t_k, k = 1 \dots M$, coarsely deformed by a homography defined by the four surrounding quad vertices. From these approximated texel instances, a rectified mean appearance of the texture element $\bar{t}_i$ can be estimated for each cluster $\mathcal{C}_i$. In the following, the term *quad* refers to a topological element of the lattice, consisting of four vertices, and *texel* defines the image part lying between those four vertices (Fig. 5.4). As the true deformation of the texels might be more complex, the lattice is only exploited to estimate a mean texel appearance and a coarse deformation. In a subsequent image-based registration step, the deformation is refined with a more detailed mesh (Fig. 5.4, Sec. 5.2.2). In the following, the lattice estimation procedure is explained in detail.

For each cluster, the lattice growth starts at a seed point $\mathbf{p}_S \in \mathcal{C}_i$ and two candidate generating vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ (Fig. 5.4). To search for a suitable seed point, a similar strategy to the one presented in [PBCL09] is used. A number of feature points and their two nearest neighbors are selected randomly, yielding sets of seed points and generating vectors $\{\mathbf{p}_S, \mathbf{t}_1, \mathbf{t}_2\}$. For each set, an affine transformation $\mathbf{A}$ is

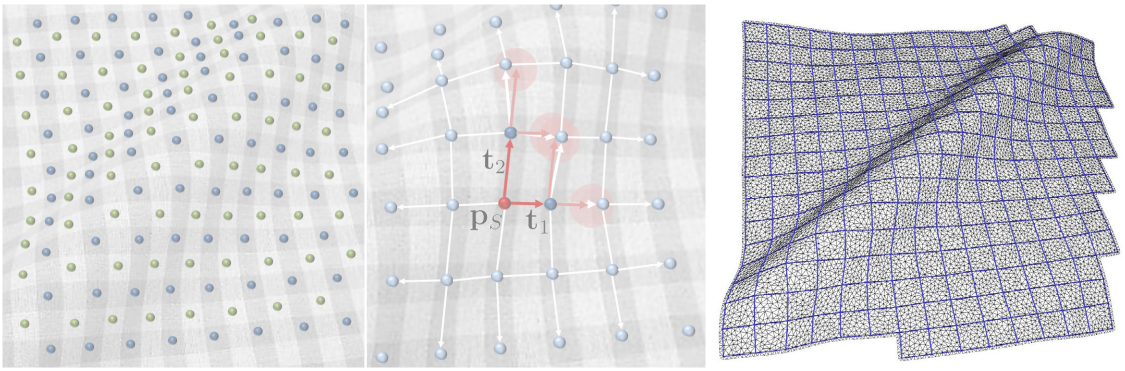


Figure 5.4.: Left: detected feature clusters marked in different colors. Center: illustration of lattice generation from feature points. Starting at a seed point $\mathbf{p}_S$ (red point) and two proposing vectors $\mathbf{t}_1$, $\mathbf{t}_2$ (red arrows), two new lattice vertices are found (dark blue points). The generating vectors are inherited by the new points (light red arrows), and their children are found in a region (light red spheres) around the proposed positions. The same procedure is done for the negative proposing vectors. Right: the mesh (black) used in the subsequent image registration step is much finer than the lattice (blue) to allow more complex deformation.

computed that maps the three points $\{\mathbf{p}_S, \mathbf{p}_S + \mathbf{t}_1, \mathbf{p}_S + \mathbf{t}_2\}$ from image space to the normalized lattice base $\{[0\ 0]^T, [0\ 1]^T, [1\ 0]^T\}$. This affine transformation is applied to all remaining cluster points, and all transformed points within some threshold of an integer lattice position are counted. The point $\mathbf{p}_S$ with the maximum number of counts is selected as seed point. This procedure favors a seed point and two generating vectors in a region where deformation is small or at least homogeneous in the local neighborhood. If user interaction is feasible, the selected seed point and its two generating vectors can be proposed to the user who can accept or discard the choice.

To grow the lattice, the four nearest cluster points in a predefined distance (depending on the mean distance between cluster points) from $\mathbf{p}_S \pm \alpha\mathbf{t}_1$ and $\mathbf{p}_S \pm \alpha\mathbf{t}_2$, $0.5 \leq \alpha \leq 2$ are determined. If such points are found, they become vertices of the lattice, and the lattice edges are defined by the vectors from the seed to these points. The vectors $\mathbf{t}_1, \mathbf{t}_2$ and $-\mathbf{t}_1, -\mathbf{t}_2$ for the new points are given by the edges between these points and their parent. The procedure starts again for all new points, until no new point can be detected.

From the established edges between cluster points, a quadrilateral lattice is generated with candidate quads that are evaluated based on texel appearance as detailed in the following paragraph. The lattice detection procedure is schematically illustrated in Fig. 5.4. If not all cluster points have been visited during the lattice growing, a new lattice growing step starts from a new seed point. Hence, the resulting lattice of one cluster can consist of several connected components, which will later grow into each other.

Texel Appearance Estimation. The following procedure is performed for each cluster $\mathcal{C}_i$. For readability reasons, the index $_i$ for the cluster will be skipped in the

following.

Each lattice quad is now associated with a candidate for a deformed texture element, which is the image region inside the quad. To reject unreliable quads, each texture element is rectified into a normalized texel coordinate system, and its intensities are normalized by subtracting the mean and dividing by the standard deviation. From these normalized texel candidates, a rectified and normalized mean texel appearance is calculated:

$$\bar{t} = \frac{1}{M} \sum_{k=1}^{M} \frac{1}{\sigma_k} (t_k - \mu_k) \ . \tag{5.3}$$

μ_k and σ_k denote the intensity mean and standard deviation of the rectified texel t_k, and M is the number of candidate quadrilaterals. The rectification to a normalized texel coordinate system is done regardless of the true texel shape to compare the appearances of the texel quads and to estimate a mean rectified appearance.

The sum of squared differences between the pixel values of each rectified and normalized texel and the mean texel appearance provides a quality measure for each texel candidate:

$$d_k = \sum_{RGB} \sum_{\mathbf{x}} \left(\bar{t}(\mathbf{x}) - \frac{1}{\sigma_k} (t_k(\mathbf{x}) - \mu_k) \right)^2 \ . \tag{5.4}$$

This similarity measure is used to remove unreliable texels and quads with a MAD-based (median of absolute differences) outlier rejection method [IH93] (App. A.1.2). The remaining quads are additionally evaluated based on a continuity measure c_k (Equ. (5.5)). A valid rectified texel of a regular texture should have similar upper and lower as well as left and right borders to produce seamless transitions during tiling. Hence, a continuity measure based on the sum of squared intensity differences between the left and right borders as well as the upper and lower borders can be defined as

$$c_k = \sum_{\mathbf{x} \in \mathcal{B}_k} \left(t_k(\mathbf{x}) - t_k(\mathbf{x}_o) \right)^2 \ , \tag{5.5}$$

where $\mathcal{B}_k$ denotes the border regions of texel t_k, and $\mathbf{x}_o$ denotes the texture border point opposite to $\mathbf{x}$. Quads with a bad continuity measure are rejected with a MAD-based outlier rejection method akin to the similarity measure [IH93] (App. A.1.2).

From the quads classified as inliers by both the similarity as well as the continuity measure, a topologically consistent quadrilateral lattice $\mathcal{L}$ is constructed, which can consist of multiple connected components. The mean texel appearance $\bar{t}$ is updated according to Equ. (5.3).

5.2.2. Warp-Based Texture Decomposition

The previous step estimated a mean rectified texel appearance $\bar{t}$ and coarse positions and deformations of texels in the image, represented by a quadrilateral lattice $\mathcal{L}$. By exploiting the assumed topology of the texture regularity, an image $\mathcal{T}$ of the undeformed texture can now be synthesized from the estimated mean texel $\bar{t}$ by

Figure 5.5.: Texel appearance estimation results.

regular tiling (Fig. 5.6). Following, the estimation of the texture deformation and the shading map is treated as a spatial and photometric image registration task, solving for a warp that registers the synthesized undeformed texture $\mathcal{T}$ onto the original image $\mathcal{I}$ both in the spatial and photometric domain. The warps are jointly estimated using the image-based warp optimization framework from Chapter 3. To be able to cope with texture discontinuities and sharp shadows, a robust estimator is used not only in the data term but also in the smoothness term.

A fine triangular mesh $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ is used as warp model to register the two images, in contrast to the quadrilateral lattice $\mathcal{L} : \{\mathbf{P}, \mathcal{Q}\}$, representing the texture topology and coarse texel positions in the image (Fig. 5.4 on page 99). The mesh-based warp is initialized with the displacements $\mathbf{d}_j = \mathbf{p}_j - \mathbf{p}_j^r$ from the regular quadrilateral lattice points $\mathbf{p}_j^r$ on $\mathcal{T}$ to the deformed lattice points $\mathbf{p}_j$ on $\mathcal{I}$. Details on warp initialization from sparse correspondences can be found in Sec. 3.2.4. In the following warp optimization, the mesh vertex positions as well as photometric parameters per vertex are updated, thereby refining the texture deformation estimation and the shading map. Fig. 5.6 shows an example result of the warp optimization and compares the original image $\mathcal{I}$ (left) with the final warped regular texture $\mathcal{W}_p\big(\mathcal{T}\big(\mathcal{W}_s(\mathbf{x})\big)\big)$ (right).

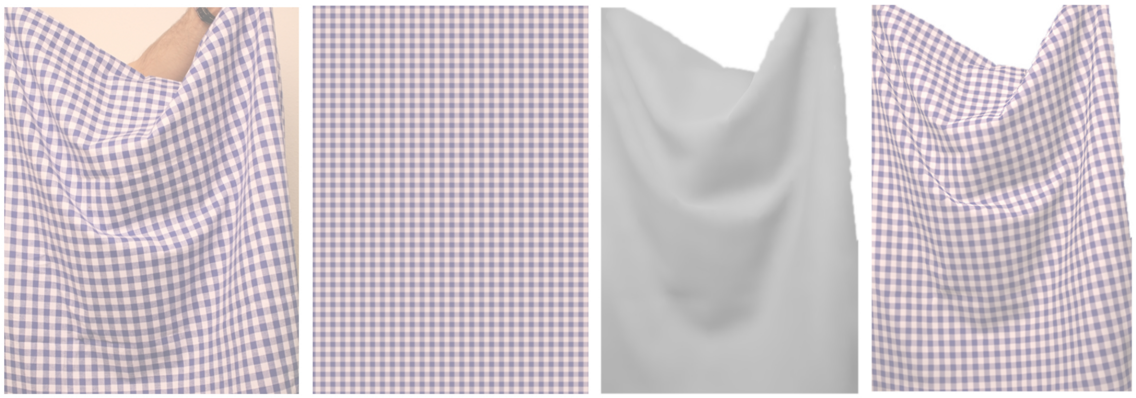


Figure 5.6.: From left to right: original image $\mathcal{I}$; estimated regular texture appearance $\mathcal{T}$; shading map after optimization; warped regular texture $\mathcal{W}_p(\mathcal{T}(\mathcal{W}_s(\mathbf{x})))$.

5.2.3. Lattice Propagation and Fusion

The lattice and the reference regular texture are grown iteratively after each optimization has reached its minimum, and a new warp optimization is started with the grown lattice/mesh. The lattice growth is performed on the undeformed regular lattice by attaching new quads to the lattice borders. New vertices are added to the regular lattice border vertices if the corresponding vertex in the deformed lattice (i) still lies inside the image, (ii) lies within the boundaries of a provided alpha mask, and (iii) does not lie in the region of another connected component of the lattice. The propagation is repeated following each warp optimization round as long as vertices fulfilling all three conditions can be found. The positions of corresponding vertices in the deformed lattice are initialized using Laplace interpolation on the lattice (compare App. A.1.3). If user interaction is feasible, the user is allowed to move newly inserted lattice vertices on the image. This is sometimes necessary at strong deformations and discontinuities, which cannot be captured by the smooth Laplace interpolation.

The lattice can consist of multiple connected components that *grow into each other*. If new vertices of one connected component of the deformed lattice would be inserted inside a quad of another connected component, the nearest vertices of this connected component are taken as vertices of the new quad, and the two connected components are fused.

5.2.4. Texture Replacement

Having processed the input image as described in the previous sections, it can now be decomposed into its intrinsic components according to Equ. (5.2), i.e. the regular texture $\mathcal{T}$, a spatial warp $\mathcal{W}_s(\mathbf{x})$ and a photometric warp $\mathcal{W}_p(\mathcal{T})$. These components can now be applied to any arbitrary new texture $\mathcal{T}_{\text{new}}$ to generate an image of this

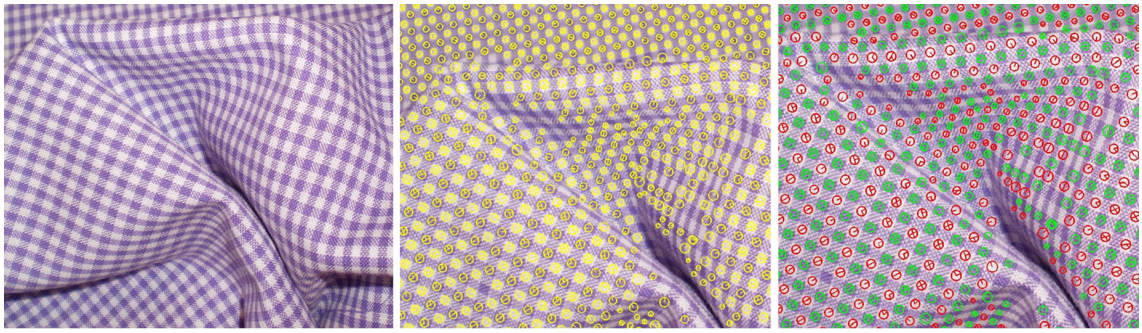


Figure 5.7.: From left to right: example for texture discontinuities at sharp creases; detected SIFT feature points (some strongly deformed texels are not detected); clustered feature points marked in different colors.

texture with the same deformation and illumination properties as the input image (see e.g. Fig. 5.9 on page 105):

$$\mathcal{I}_{\text{new}}(\mathbf{x}) = \mathcal{W}_p\big(\mathcal{T}_{\text{new}}\big(\mathcal{W}_s(\mathbf{x})\big)\big) \ . \tag{5.6}$$

A texture $\mathcal{T}_{\text{new}}$ of the correct size can be synthesized from a small texture sample using the texture synthesis method described in [RHE11].

Saturation, Specularities and Gamma Correction. Of course, a linear shading model, such as the photometric warp model used in this thesis, is not strictly true, especially in case of saturation and specularities. Saturation or specularities are treated by thresholding the resulting values in $\mathcal{I}_{\text{new}}$. This procedure showed to be perceptionally convincing. Gamma correction needs not be handled separately, because the photometric model is multiplicative (Equ. (3.48)), which means that the Gamma correction factor is also applied to the multiplicative photometric parameters, as illustrated in the following example:

> Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two spatially registered images where Gamma correction has been applied. Let $\tilde{\mathcal{I}}_1 = \mathcal{I}_1^{1/\gamma}$ and $\tilde{\mathcal{I}}_2 = \mathcal{I}_2^{1/\gamma}$ be the same images without Gamma correction. A multiplicative shading model yields $\mathcal{I}_2(\mathbf{x}) = \alpha \cdot \mathcal{I}_1(\mathbf{x})$ with $\alpha = \mathcal{W}_p(\mathbf{x})$. Hence, $\tilde{\mathcal{I}}_2(\mathbf{x}) = \beta \cdot \tilde{\mathcal{I}}_1(\mathbf{x})$ with $\beta = \alpha^{1/\gamma}$.

5.3. Discussion and Results

The presented texture decomposition approach has been tested on several images of cloth textures, and the results of the different processing steps are discussed in the following. To cope with sharp texture discontinuities, user interaction was allowed in the presented examples to discard wrongly detected quads after the initial lattice estimation and to move newly inserted lattice vertices during lattice growth.

Texel Detection and Lattice Estimation. In all presented results, SIFT features have been used [Low03] to detect and cluster repeating elements in the image.

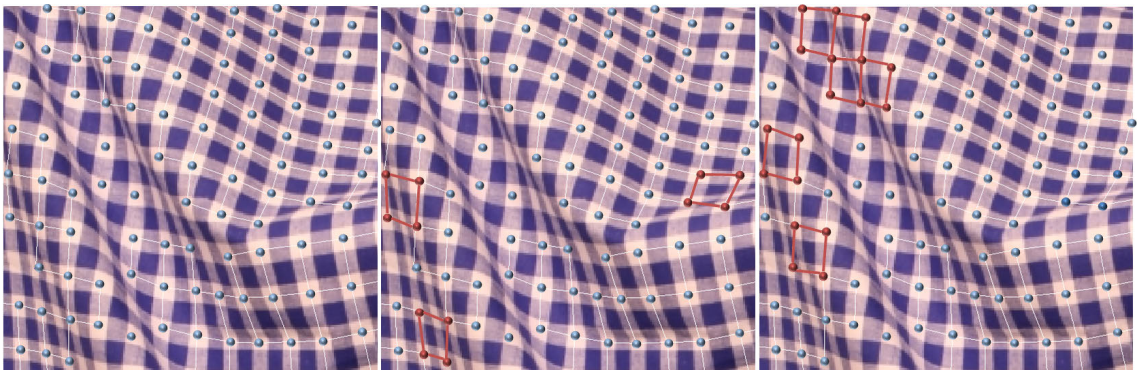


Figure 5.8.: Examples for wrongly detected quads. From left to right: detected quads; wrong quads discarded by the appearance-based outlier detection method marked in red; wrong quads at discontinuities discarded by the user marked in red.

Although the SIFT descriptor is not invariant against affine or projective transformations, and hence may miss some strongly distorted structures (Fig. 5.7), it detected enough feature points to estimate an initial texel appearance and lattice structure. Missed texels are later filled in during warp optimization and lattice propagation. The lattice estimation algorithm might propagate over sharp creases with texture discontinuities. Most of these cases were detected by the MAD-based outlier detection method based on the texel similarity measure and the continuity measure with a threshold of $\kappa = 3.5$ (Equ. (5.4)-(5.5)). In some cases, a detection is nearly impossible, e.g. if the crease is exactly at the border of two texels (see Fig. 5.8). Therefore, the estimated deformed lattice is displayed to the user, who can select wrongly detected texels or quads to be removed. Missed texel positions are later filled in topologically correct during the lattice growth, and their positions are refined during the warp estimation. Fig. 5.8 shows examples for wrong quads discarded by the appearance-based measures and wrong quads that needed to be removed by the user. Results for the texel appearance estimation are depicted in Fig. 5.5 on page 101 for a variety of regular cloth patterns.

Warp Optimization and Retexturing. For joint spatial and photometric warp optimization between the synthetically generated appearance of the regular texture and the input image, the image-based warp optimization approach of Chapter 3 is used with a robust estimator in the data term as well as the smoothness term. The robust estimator in the smoothness term allows the modeling of texture and shading discontinuities at sharp creases and wrinkles. This can be seen e.g. in Fig. 5.9. The quality of warp estimation and retexturing results is best evaluated visually. Fig. 5.6 on page 102 shows the decomposition result of an original image $\mathcal{I}$ (left) into the undeformed regular texture $\mathcal{T}$, a deformation field $\mathcal{W}_s(\mathbf{x})$ and a shading map $\mathcal{W}_p(\mathcal{T})$. The right most image shows a synthetic retexturing result with the estimated regular texture appearance, i.e. a reproduction of the original image. The original image (left) and the synthetic image (right) can now be compared to evaluate the retexturing result. Although marginal differences between the images are noticeable when compared directly, the visual appearance of the synthetic result

Figure 5.9.: Retexturing results for the image depicted in Fig. 5.6 on page 102. The undeformed textures for retexturing were synthesized from the small samples in the lowest row using the method described in [RHE11].

is still realistic and plausible. Retexturing results with synthetic textures for this image are depicted in Fig. 5.9. Further retexturing results are presented in Fig. 5.10-5.11 on the following pages. In the augmented results, the spatial deformation and shading conditions of the original texture are maintained such that the object seems to truly exhibit the new synthetic texture. The estimation of a photometric warp is essential for a convincing and realistic appearance of the augmented image. This is demonstrated by Fig. 5.12 on page 107, which directly compares retexturing results with and without accounting for shading. Without the addition of the shading properties extracted from the original image, the synthetic texture does not appear to be integrated into the scene, whereas a photometric warp increases the perception of spatial relations between the real world and the new texture. Note that the spatial deformation applied to the new texture is the same in each image pair.

Figure 5.10.: Retexturing of a dress. The first two images in the top row show the original image and the shading map. The undeformed textures were synthesized from the small samples in the lowest row using the method described in [RHE11].

5.4. Chapter Summary

This chapter has presented an image-based retexturing method, which formulates the problem of texture deformation and shading estimation as a non-rigid image registration task both in the spatial as well as in the photometric domain. It exploits the joint spatial and photometric warp optimization framework from Chapter 3 to extract local deformation and shading properties from an image by registering a synthetic reference image showing the undeformed and uniformly lit texture. This reference image is estimated from the input image, assuming that it shows a texture

Figure 5.11.: Retexturing pillows. The two leftmost images depict the original input images and the estimated shading maps.

Figure 5.12.: Comparison of retexturing results with and without accounting for shading (the spatial deformation is the same in the left and right images of each example). These examples demonstrate how important the estimation of a photometric warp is for realistic retexturing.

of a near-regular type. Various retexturing results present the effectiveness of the presented approach and the benefit for the realistic appearance of the synthetic result added by the photometric warp component.

In addition to retexturing, the proposed texture analysis approach has been applied to estimate the 3D shape of a surface in a template-free shape-from-texture approach [HSE11b]. Under the assumptions that the texture is constructed of one or more repeating elements and that the texture element is small enough to be modeled as a planar patch, shape-from-texture methods compute the 3D shape of a textured surface by exploiting texture distortion as a cue for shape. In [HSE11b], the 3D

shape is calculated from homographies between the texture elements, assuming a perspective camera model. As shape-from-texture is out of the scope of this thesis, the interested reader is referred to [HSE11b] for more information.

6. Conclusions

This dissertation has presented novel image-based approaches to photo-realistically render, synthesize and modify images of complex objects, concentrating on the example of clothing. The basic concept of the proposed methods is to rely on the use of real images instead of reconstructing and simulating the underlying scene properties. This approach exploits the fact that very characteristic details, which are difficult to synthesize, such as wrinkling, texture deformation and shading, can be captured by the images. These details are extracted from the images as mesh-based warps both in the spatial as well as photometric domain such that they can serve as appearance examples to guide a complex animation or retexturing process.

The proposed methods shift computational complexity from the rendering phase to an a-priori training phase and rendering amounts to warping a set of images.

Contributions. Three main contributions have been presented.

A new image-based rendering approach for articulated clothing has been introduced. In contrast to classical image-based rendering approaches, which are restricted to viewpoint interpolation, a body pose-dependent method has been developed. This method generates new images based on pose parameters by interpolating and merging images of clothes from a database of examples. A coarse geometric model allows animation and view interpolation, while small details as well as complex shading and reflection properties are modeled by pose-dependent appearance examples in the database. Correspondences between the images are represented as mesh-based warps, both in the spatial and intensity domain. These warps capture fine pose-dependent texture deformation and shading information. For rendering, the images and warps are interpolated in pose-space, i.e. the space of body poses, using scattered data interpolation. The concept of interpolating sample body poses has been transferred and modified from example-based animation methods to image-based rendering. Related issues, such as blending and photo-consistency, have been thoroughly discussed and analyzed. The high dimensionality of the pose-space has been addressed by splitting up the pose-space into subspaces of body parts. This reduces the dimensionality of the interpolation domain as well as the number of required examples and allows for a modeling of different body parts from different database images. Results have been discussed based on various pose interpolation as well as extrapolation examples. Generally, the proposed method is not limited to the visualization of clothing but can rather be applied to any articulated object with pose-dependent appearance, e.g. realistic avatar rendering.

To allow for appearance modification in image-based methods, an image-based approach to retexturing an image or video has been proposed. In this method, texture

deformation and shading are extracted from the input image or video in a warp-based approach, i.e. by registering the input image onto an appropriate reference image of the undeformed and uniformly lit texture, both in the spatial as well as in the photometric domain. Because such a reference image is not always available, a texture analysis method for near-regular textures has been introduced in this context, which estimates the appearance of the associated regular texture, thereby synthetically generating a reference image. Texture deformation and shading properties are modeled and extracted as spatial and photometric warps, which can, once extracted, be applied to any new texture to be rendered into the original image or video. Various retexturing examples have been presented, demonstrating how important the preservation of shading and hence the extraction of photometric warps are for a visually correct retexturing result.

Both presented approaches build on the concept of joint spatial and photometric mesh-based warps between images, and this dissertation has presented a new framework for the joint estimation of these warps, based on a relaxed brightness constancy assumption. The presented approach enables a compact extraction of local texture deformation and shading differences between two images. Various applications of such warps have been presented throughout the thesis. Experiments have shown that incorporating a photometric warp into the cost function, not only allows extraction of texture deformation and shading, but also the estimation of spatial warps based on image intensity becomes more robust against lighting differences between the images.

Applications and Outlook. The applications of the presented methods are manifold. The main targeted application is the visualization of clothes in augmented reality applications, such as virtual try-on, in which a user can see himself with virtual clothes while moving freely in front of the system. Based on the estimated pose of the user, an image of the selected piece of clothing can be synthesized from a stored database and mapped onto the image of the user using the methods of Chapter 4. To realize such a real-time Virtual Mirror system, the proposed image-based animation method needs to be combined with a real-time body tracking system. In such a scenario, the pose parameterization, i.e. skeleton configuration, of both systems should be consistent. The synthesized piece of clothing will be rendered onto the body of the user, which might be of a different shape than the person wearing the piece of clothing during the training phase. Small shape differences can be compensated by warping the rendered piece of clothing onto the silhouette of the user. For larger shape differences, an additional shape parameter in the parameterization of the database images could be investigated.

Having a virtual try-on application in mind, it will be promising to combine the two presented approaches of Chapter 4 and Chapter 5 by including retexturing information, i.e. texture deformation and shading information, into the proposed database representation of Chapter 4. This will not only allow animation of a pre-captured piece of clothing but also to change and modify its appearance during rendering. This would enhance the database representation in Fig. 4.1 on page 57 by an additional partition of the appearance information into texture deformation,

shading and texture albedo. The texture analysis process of Chapter 5 needs to be performed only once for one example image in the database. For all other database images, the information can be propagated using the extracted warps between the images.

Finally, not being limited to the visualization of clothing, the presented methods will also be especially useful for applications in movies or films. New animations could be synthesized from pre-captured poses of actors in a post-processing stage, thereby enabling a larger flexibility in all processing steps. Similarly, image-based animation and rendering methods will be of benefit in computer games to create photo-realistic and interactive characters.

Summing up, exploiting images and warp-based texture deformation and shading representation has been shown to be a promising concept for photo-realistic rendering and retexturing of clothing with many potential applications in computer graphics and augmented reality.

A. Appendix

A.1. Mathematical Derivations and Definitions

A.1.1. Camera Model

A camera model consists of a set of extrinsic parameters, defining the position and orientation of the camera in the world coordinate system, and a set of intrinsic parameters, defining how the camera forms an image. The intrinsic parameters include the focal length in pixels f_x, f_y and the principal point p_x, p_y. These parameters build the intrinsic camera matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} . \tag{A.1}$$

The extrinsic camera parameters consist of a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. The full camera projection matrix is given by

$$\mathbf{P} = \mathbf{K} \left[\mathbf{R} \ \mathbf{t} \right] . \tag{A.2}$$

This matrix projects a 3D point $\mathbf{p}$ into the image by

$$\mathbf{x}^h = \mathbf{K} \left[\mathbf{R} \ \mathbf{t} \right] \cdot \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} . \tag{A.3}$$

where $\mathbf{x}^h$ denotes a 2D pixel in homogeneous coordinates, and the corresponding Cartesian point $\mathbf{x}$ is found by

$$\mathbf{x} = \frac{1}{x_3^h} \begin{bmatrix} x_1^h \\ x_2^h \end{bmatrix} . \tag{A.4}$$

In this thesis, the projection of a 3D point $\mathbf{p}$ to a 2D image point $\mathbf{x}$ by the camera matrix $\mathbf{P}$ is denoted by $\mathbf{x} = \mathcal{P}_{\mathbf{P}}(\mathbf{p})$.

A.1.2. MAD-Based Outlier Rejection

For successful outlier rejection, a resistant score is required that is not significantly influenced by the outliers. The mean and the variance are shifted by the presence of outliers in the data. Hence, many outliers are not detected with mean-based scores. The median and the median of absolute deviations (MAD) are more robust against outliers [IH93]. Let $\mathbf{x} = \{x_1, \ldots, x_N\}$ denote a set of N data points and $\tilde{x}$ denote the median of the data. The MAD is defined as:

$$\mathrm{MAD}(\mathbf{x}) = \mathrm{median}\,|\mathbf{x} - \tilde{x}| \ ,$$

where $|.|$ denotes the absolute value of a scalar. The MAD score for a datum x_i is defined as

$$M_i = \left| \frac{x_i - \tilde{x}}{MAD(\mathbf{x})} \right| \ .$$

Any observation is rejected as an outlier if $M_i > \kappa_{\mathrm{MAD}}$ where κ_{MAD} is the maximum permissible MAD score. For large N and Normal distribution, a value of $\kappa_{\mathrm{MAD}} = 3.5$ is suggested in the literature [IH93].

A.1.3. Laplace Interpolation

General Formulation. Suppose known scattered data $f^*(\mathbf{x}_i) = f_i^*$ in 2D at supporting positions $\mathbf{x}_i \in \mathbb{R}^2$ and the aim is to interpolate a smooth function $f(\mathbf{x})$ from these control points. Let $\partial\Omega$ denote the magnitude of all supporting positions (often called the *boundary*) and Ω be the region of all unknown positions: $\Omega \bigcap \partial\Omega = \emptyset$. The idea of Laplace interpolation is to apply a constraint on the smoothness of an interpolating function f that minimizes the integrated square of the gradient of the function while fulfilling the boundary conditions [PGB03]:

$$\min_f \int_\Omega |\nabla f|^2 \, d\Omega \ \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \ . \tag{A.5}$$

Discrete Formulation. A discrete approximation of Equ. (A.5) on a 2D grid or mesh yields the following quadratic optimization problem:

$$\min_f \sum_{\mathbf{x}_i \in \Omega} \left(f_i - \frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} f_n \right)^2 \ \text{ with } \ f_i = f_i^* \ , \forall \mathbf{x}_i \in \partial\Omega \ , \tag{A.6}$$

with $f_i = f(\mathbf{x}_i)$. $\mathcal{N}_i$ denotes the neighborhood of the data point $\mathbf{x}_i$ on the regular grid or mesh. The solution satisfies a linear equation system with equations

$$f_i - \frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} f_n = 0 \ \forall \mathbf{x}_i \in \Omega \tag{A.7}$$

and

$$f_i = f_i^* \ \forall \mathbf{x}_i \in \partial\Omega \ . \tag{A.8}$$

Let $\mathbf{L}_\Omega$ denote the Laplacian matrix of the data in the region Ω with entries

$$l_{ij} = \begin{cases} -1 & i = j \\ \frac{1}{|\mathcal{N}_i|} & \mathbf{x}_j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{A.9}$$

per data point $\mathbf{x}_i \in \Omega$. Equ. (A.6) can then be written as

$$\hat{\mathbf{f}} = \min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_\Omega \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{f}^* \\ \mathbf{0} \end{bmatrix} \right\|^2 \ , \tag{A.10}$$

where $\mathbf{f}^*$ is the vector of all values $f_i^* = f^*(\mathbf{x}_i)$ at supporting positions $\mathbf{x}_i \in \partial\Omega$. The equation system in Equ. (A.10) has N equations and N unknowns and can be solved uniquely.

Discrete Approximation. To calculate an approximative but smoother result, Laplacian smoothing may be applied in the complete region $\mathcal{R} = \Omega \bigcup \partial\Omega$. Let $\mathbf{L}_\mathcal{R}$ denote the Laplacian matrix of the data in region $\mathcal{R}$. An approximative but smooth result to the given problem is found by minimizing the following least-squares problem:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{D} \\ \lambda \mathbf{L}_\mathcal{R} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{f}^* \\ \mathbf{0} \end{bmatrix} \right\|^2 \ . \tag{A.11}$$

λ regularizes the smoothness against fitting to the data term and $\mathbf{D} = [\mathbf{I}\ \mathbf{0}]$.

For meshes, the position of known data points do not necessarily coincide with vertex positions but rather lie between the vertices. In this case, they still have a unique position in the mesh $\mathcal{M} : \{\mathbf{V}, \mathcal{F}\}$ defined by the barycentric coordinates with respect to the surrounding triangle:

$$\mathbf{x}_i = \sum_{l=1}^{3} \beta_l \mathbf{v}_l \ . \tag{A.12}$$

Here, $\mathbf{v}_l$ are the three vertices of the surrounding triangle and β_l are the corresponding barycentric coordinates. The data at unknown vertex positions can now be interpolated by solving the following equation system:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\| \begin{bmatrix} \mathbf{D} \\ \lambda \mathbf{L} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{f}^* \\ \mathbf{0} \end{bmatrix} \right\|^2 \ . \tag{A.13}$$

$\mathbf{D}$ is a data matrix with one row vector $\mathbf{p}^i$ per data point $\mathbf{x}_i \in \Omega$ with entries:

$$\mathbf{p}_j^i = \begin{cases} \beta_l & \text{if } \mathbf{v}_j \text{ is the } l^{\text{th}} \text{ vertex in the triangle surrounding } \mathbf{x}_i \in \Omega \\ 0 & \text{otherwise} \end{cases} \ . \tag{A.14}$$

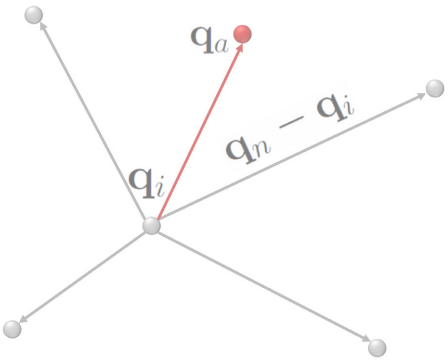


Figure A.1.: A pose p_a, for which a warp can be interpolated from a sample pose p_i, is constrained by its position plus the positions of its neighbors in the pose-graph, i.e. the direction of the stored warps.

A.1.4. Pose-Space Constraints

With the interpolation scheme introduced in Sec. 4.2.1, a pose p_a, for which a warp can be interpolated from a sample pose p_i, is constrained by its position plus the positions of its neighbors in the pose-graph, i.e. the direction of the stored warps (see Fig. A.1)

$$\begin{aligned}
\mathbf{q}_a &= \mathbf{q}_i + \sum_{p_n \in \mathcal{N}_i} w_n (\mathbf{q}_n - \mathbf{q}_i) \\
&= \mathbf{q}_i + \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_n - \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_i \\
&= \mathbf{q}_i + \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_n - \mathbf{q}_i \cdot \sum_{p_n \in \mathcal{N}_i} w_n \\
&= \sum_{p_n \in \mathcal{N}_i} w_n \mathbf{q}_n \text{ with } \sum_{p_n \in \mathcal{N}_i} w_n = 1 \quad .
\end{aligned} \tag{A.15}$$

A.2. Datasets

Middlebury Cloth Dataset. The Cloth1-4 stereo datasets used in Sec. 3.3 for the evaluation of the warp optimization framework were taken from the Middlebury 2006 stereo dataset[1] [HS07]. In the experiments, views 1 and 5 were used in full resolution (1252×1110 for Cloth1 and Cloth3 and 1300×1110 for Cloth2 and Cloth4), for which disparity maps are provided. Furthermore, the dataset provides three different illuminations (1-3) with three different exposures (0-2). Besides stereo pairs with the same illumination, pairs captured under different illuminations and exposures (Illum1 and Illum2) have been used for the evaluation. An overview of the

[1] http://vision.middlebury.edu/stereo/data/scenes2006/, downloaded on February 19,2013

Name	View 1	View 5
Cloth1 Illum1	Illum 1 Exp 2	Illum 2 Exp 2
Cloth1 Illum2	Illum 3 Exp 2	Illum 1 Exp 2
Cloth2 Illum1	Illum 3 Exp 2	Illum 1 Exp 1
Cloth2 Illum2	Illum 2 Exp 2	Illum 3 Exp 2
Cloth3 Illum1	Illum 1 Exp 2	Illum 3 Exp 1
Cloth3 Illum2	Illum 2 Exp 1	Illum 3 Exp 1
Cloth4 Illum1	Illum 2 Exp 2	Illum 3 Exp 1
Cloth4 Illum2	Illum 1 Exp 1	Illum 1 Exp 2

Table A.1.: Illumination combination for the Middlebury Cloth datasets used in Sec. 3.3. The images are depicted in Fig. A.2.

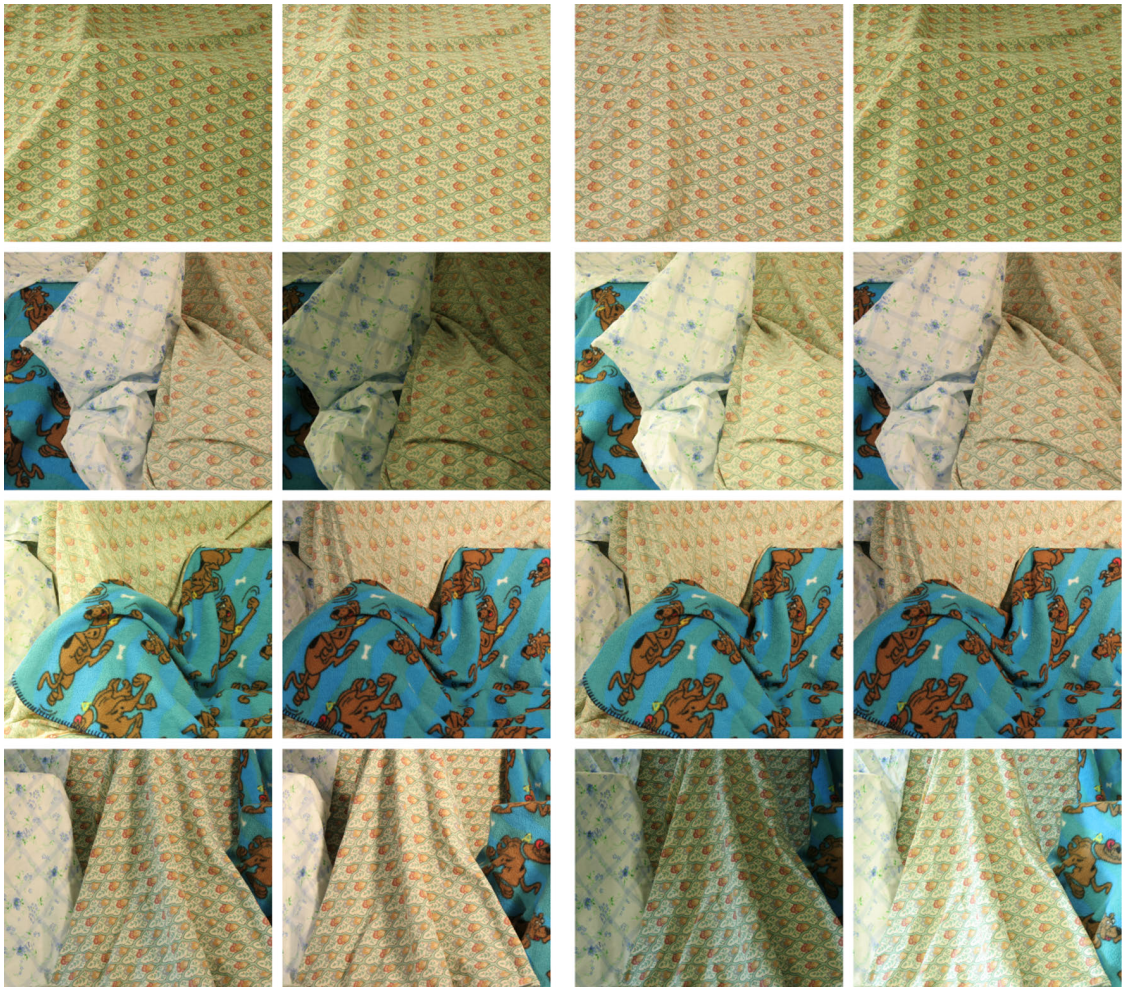


Figure A.2.: Image pairs with different illuminations and exposures from the Middlebury Cloth dataset [HS07] used in Sec. 3.3, as listed in Tab. A.1. From top to bottom: Cloth1-4. Left pair: Illum1. Right pair: Illum2

combined illuminations and exposures for the different datasets is given in Tab. A.1, and the image pairs are depicted in Fig. A.2.

Name	Resolution	fps	Length	Note
Bedsheet	720×576	10	67	bed_sheet dataset used in [SMNLF08]
Cushion	720×576	10	70	cushion dataset used in [SHF07]
Flowers	768×1024	25	350	captured with a Flea2 XGA-Firewire camera
Paper1	720×576	25	100	data_3 dataset used in [GBBS10]
Paper2	720×576	10	70	paper_bending dataset used in [SHF07]
Paper3	768×1024	25	100	captured with a Flea2 XGA-Firewire camera
Shirt	640×480	10	245	tshirt_gt dataset used in [VSFU12]
Testpattern	768×1024	25	500	captured with a Flea2 XGA-Firewire camera

Table A.2.: Tracking dataset used in Sec. 3.3. Example frames of all sequences are depicted in Fig. A.3. The datasets Bedsheet, Cushion, Shirt from [SHF07, SMNLF08, VSFU12] are available under http://cvlab.epfl.ch/data/dsr.

Name	Poses	Views	Note
Jeans	22	9	Tight fitting long trousers, denim material
Jacket	40	8	Moderately tight fitting jacket with long sleeves, denim material
Shirt1	11	8	Tight fitting shirt with short sleeves, soft material
Shirt2	20	7	Tight fitting shirt with long sleeves, soft material
Blouse	11	3	Loose fitting blouse with long sleeves, soft material

Table A.3.: Clothing datasets used in Chapter 4.

Tracking Dataset. Tab. A.2 lists the video sequences that were used for the deformable tracking evaluation in Sec. 3.3, with resolution, number of frames per second, total length in frames, and a note on the origin of the video sequence. Example frames are depicted in Fig. A.3. Fig. A.4 plots the mean intensity RMSE over all eight sequences for all three tested warping models (s, sp, spc, see Tab. 3.1 on page 47).

Clothing Datasets for Pose-Space Image-Based Rendering. Tab. A.3 lists the clothing databases used and analyzed in Chapter 4 .

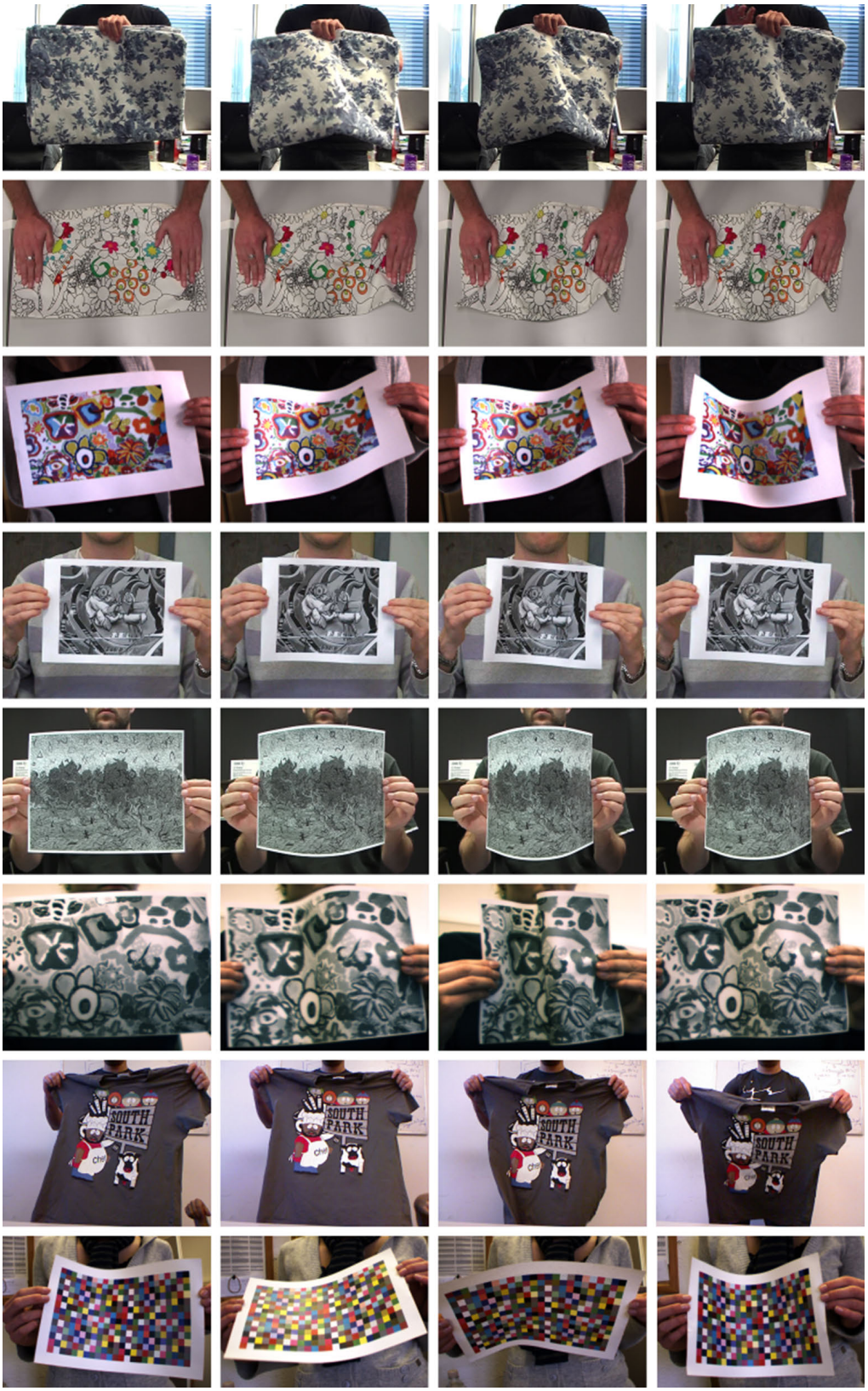


Figure A.3.: Example frames of the eight video sequences used for tracking evaluation in the same order as in Tab. A.2.

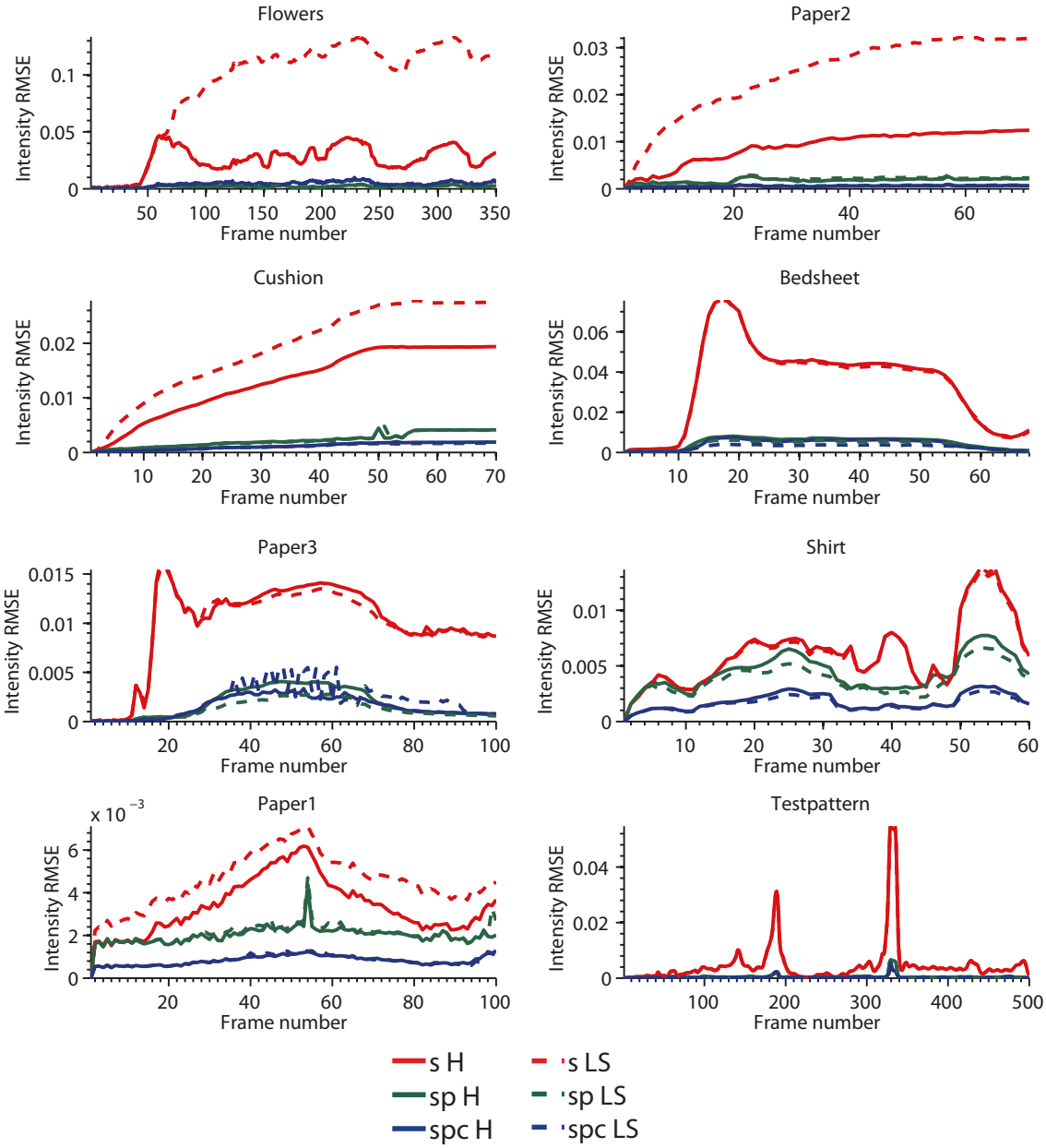


Figure A.4.: Plots of the RMSE between the synthetic frame $\hat{\mathcal{I}}_j$ and the original current frame $\mathcal{I}_j$ for the three different warping models (s, sp, spc, see Tab. 3.1 on page 47) for the video sequences listed in Tab. A.2. For comparison, the results achieved with least-squares optimization (dashed) and robust optimization (solid) are depicted.

References

[ABT05] R. Aster, B. Borchers, and C. Thurber. *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, Jan. 2005. 26, 28, 51

[ACP02] B. Allen, B. Curless, and Z. Popović. Articulated Body Deformation from Range Scan Data. *ACM Trans. Graph. (Proc. SIGGRPAH)*, 21(3):612–619, July 2002. 17, 55, 59, 68, 69, 72

[AJ09] M. Aigner and B. Jüttler. Distance Regression by Gauss-Newton-Type Methods and Iteratively Re-Weighted Least-Squares. *Computing*, 86(2):73–87, Oct. 2009. 26

[BA93] M. J. Black and P. Anandan. A Framework for Robust Estimation of Optical Flow. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 231–236. IEEE Computer Society, May 1993. 10, 11

[BAHH92] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 237–252. Springer, May 1992. 11

[Bar08] A. Bartoli. Groupwise Geometric and Photometric Direct Image Registration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12):1098–2108, Dec. 2008. 12

[BBM+01] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured Lumigraph Rendering. In *Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 425–432. ACM, Aug. 2001. 3, 13, 69

[BBPP10] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured Video-Based Rendering: Interactive Exploration of Casually Captured Videos. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29(4):87:1–87:11, July 2010. 14

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 25–36. Springer, May 2004. 10, 11, 12, 34

[BETG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. 10

[BFB94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of Optical Flow Techniques. *Int. Journ. of Computer Vision*, 12(1):43–77, Feb. 1994. 9, 11

[BKP+10] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. A K Peters, Jan. 2010. 32, 33

[BM92] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992. 65

[BM04] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. Journ. Comput. Vision*, 56(3):221–255, March 2004. 9

[BM11] T. Brox and J. Malik. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):500–513, March 2011. 11

[BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of Clothing with Folds and Wrinkles. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 28–36. Eurographics Association, July 2003. 55

[Boo89] F. L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, June 1989. 9

[BP07] I. Baran and J. Popović. Automatic Rigging and Animation of 3D Characters. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):72:1–72:8, July 2007. 63

[BPS+08] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless Garment Capture. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3):99:1–99:9, Aug. 2008. 3, 9, 18

[BRB09] D. Bradley, G. Roth, and P. Bose. Augmented Reality on Cloth with Realistic Illumination. *Mach. Vision Appl.*, 20(2):85–92, Jan. 2009. 18

[BTH+03] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popović, and S. M. Seitz. Estimating Cloth Simulation Parameters from Video. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 37–51. Eurographics Association, July 2003. 3

[BZ04] A. Bartoli and A. Zisserman. Direct Estimation of Non-Rigid Registrations. In *Proc. British Machine Vision Conf. (BMVC)*, pages 221–231. BMVA, Sept. 2004. 9, 10, 24

[CK05] K.-J. Choi and H.-S. Ko. Research Problems in Clothing Simulation. *Comput. Aided Des.*, 37(6):585–592, May 2005. 3

[CLSMT01] F. Cordier, W. Lee, H. W. Seo, and N. Magnenat-Thalmann. Virtual Try-On on the Web. In *Proc. Virtual Reality Int. Conf. (VRIC)*, May 2001. 21

[CM02] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002. 97, 98

[CMU] CMU NRT Database. http://vivid.cse.psu.edu/texturedb/gallery/. 20

[CSMT03] F. Cordier, H. Seo, and N. Magnenat-Thalmann. Made-to-Measure Technologies for an Online Clothing Store. *IEEE Comput. Graph. Appl.*, 23(1):38–48, Jan. 2003. 21

[CSN07] S.C. Chan, H.-Y. Shum, and K.-T. Ng. Image-Based Rendering and Synthesis. *IEEE Signal Processing Magazine*, 24(6):22–33, Nov. 2007. 13

[CTMS03] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-Viewpoint Video of Human Actors. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 22(3):569–577, July 2003. 14

[CYJ02] D. Cobzas, K. Yerex, and M. Jägersand. Dynamic Textures for Image-Based Rendering of Fine-Scale 3D Structure and Animation of Non-Rigid Motion. *Comput. Graph. Forum (Proc. Eurographics)*, 21(3):493–502, Sept. 2002. 15

[CZN+11] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao. Learning a 3D Human Pose Distance Metric from Geometric Pose Descriptor. *IEEE Trans. Vis. Comput. Graph.*, 17(11):1676–1689, Nov. 2011. 59

[Dar98] T. Darrell. Example Based Image Synthesis of Articulated Figures. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 768–774. MIT Press, Dec. 1998. 15

[DLD12] A. Davis, M. Levoy, and F. Durand. Unstructured Light Fields. *Comput. Graph. Forum (Proc. Eurographics)*, 31(2):305–314, May 2012. 3, 13

[DMC+12] M. Dellepiane, R. Marroquim, M. Callieri, P. Cignoni, and R. Scopigno. Flow-Based Local Optimization for Image-to-Geometry Projection. *IEEE Trans. Vis. Comp. Graph.*, 18(3):463–474, March 2012. 13

[DTE+04] A. Divivier, R. Trieb, A. Ebert, H. Hagen, C. Gross, A. Fuhrmann, V. Luckas, J. L. Encarnacao, E. Lirkchdöfer, S. Kimmerle, M. Keckeisen, M. Wacker, W. Strasser, R. Sarlette, and R. Klein. Virtual Try-On Topics in Realistic, Individualized Dressing in Virtual Reality. In *Proc. Virtual and Augmented Reality Status Conference*, pages 1–17. BMBF, Feb. 2004. 5, 21

[DTM96] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach. In *Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 11–20. ACM, Aug. 1996. 13

[EDM+08] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *Comput. Graph. Forum (Proc. Eurographics)*, 27(2):409–418, April 2008. 13

[EFR08] P. Eisert, P. Fechteler, and J. Rurainsky. 3-D Tracking of Shoes for Virtual Mirror Applications. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6. IEEE Computer Society, June 2008. 21

[EG97] P. Eisert and B. Girod. Model-Based 3D-Motion Estimation with Illumination Compensation. In *Proc. Int. Conf. on Image Processing and its Applications*, pages 194–198, July 1997. 12

[EG02] P. Eisert and B. Girod. Model-Based Enhancement of Lighting Conditions in Image Sequences. In *Proc. SPIE Visual Communications and Image Processing (VCIP)*, pages 260–267. SPIE, Jan. 2002. 12

[EJH10] P. Eisert, C. Jacquemin, and A. Hilsmann. Virtual Jewel Rendering for Augmented Reality Environments. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 1813–1816. IEEE Computer Society, Sept. 2010. 13

[ER06] P. Eisert and J. Rurainsky. Geometry-Assisted Image-Based Rendering for Facial Analysis and Synthesis. *Sig. Proc.: Image Comm.*, 21(6):493–505, July 2006. 13

[ES03] M. Emori and H. Saito. Texture Overlay onto Deformable Surface Using Geometric Transformation. In *Proc. Int. Conf. Artificial Reality and Tele-Existence (ICAT)*, pages 58–65. Virtual Reality Society of Japan, Dec. 2003. 22

[ES05] J. Ehara and H. Saito. Texture Overlay onto Deformable Surface for Virtual Clothing. In *Proc. Int. Conf. Artificial Reality and Tele-Existence (ICAT)*, pages 172–179. ACM, Dec. 2005. 22

[ES06] J. Ehara and H. Saito. Texture Overlay for Virtual Clothing Based on PCA of Silhouettes. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 139–142. IEEE Computer Society, Oct. 2006. 15

[FB81] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and

Automated Cartography. *Commun. of the ACM*, 24(6):381–395, June 1981. 10

[FH04] H. Fang and J. C. Hart. Textureshop: Texture Synthesis as a Photograph Editing Tool. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):354–359, Aug. 2004. 18, 19

[FHE12] P. Fechteler, A. Hilsmann, and P. Eisert. Kinematic ICP for Articulated Template Fitting. In *Proc. Int. Workshop on Vision, Modeling, and Visualization Workshop (VMV)*, pages 215–216. Eurographics Association, Nov. 2012. 63

[For02] D. A. Forsyth. Shape from Texture without Boundaries. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 225–239. Springer, May 2002. 19, 20

[GB97] P. Golland and A. M. Bruckstein. Motion from Color. *Comput. Vis. Image Underst.*, 68(3):346–362, Dec. 1997. 11

[GBBS10] V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct Estimation of Non-Rigid Registrations with Image-Based Self-Occlusion Reasoning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):87–104, Jan. 2010. 9, 12, 18, 19, 30, 118

[GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 43–54. ACM, Aug. 1996. 13

[GHF+07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient Simulation of Inextensible Cloth. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):49:1–49:8, July 2007. 3

[GHK+10] M. Germann, A. Hornung, R. Keiser, R. Ziegler, S. Würmlin, and M. Gross. Articulated Billboards for Video-Based Rendering. *Comput. Graphics Forum (Proc. Eurographics)*, 29(2):585–594, May 2010. 14

[GKT+08] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense Image Registration through MRFs and Efficient Linear Programming. *Medical Image Snalysis*, 12(6):731–741, December 2008. 12

[GN87] M. A. Gennert and S. Negahdaripour. Relaxing the Brightness Constancy Assumption in Computing Optical Flow. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, June 1987. 12

[GP10] L. J. Grady and J. R. Polimeni. *Discrete Calculus–Applied Analysis on Graphs for Computational Science*. Springer, Aug. 2010. 34

[GRH+12] P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. J. Black. DRAPE: DRessing Any PErson. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(3):35:1–35:10, Aug. 2012. 18

[GS86] B. Grünbaum and G. C. Shephard. *Tilings and Patterns: An Introduction*. W. H. Freeman & Co., New York, NY, USA, Sept. 1986. 19, 94, 95

[GSPJ08] Y. Guo, H. Sun, Q. Peng, and Z. Jiang. Mesh-Guided Optimized Retexturing for Image and Video. *IEEE Trans. Vis. Comput. Graph.*, 14(2):426–439, March 2008. 18, 19

[GWO+10] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless Montage for Texturing Models. *Comput. Graph. Forum (Proc. Eurographics)*, 29(2):479–486, May 2010. 13

[HDK07] A. Hornung, E. Dekkers, and L. Kobbelt. Character Animation from 2D Pictures and 3D Motion Data. *ACM Trans. Graph.*, 26(1):1–9, Jan. 2007. 15

[HE08] A. Hilsmann and P. Eisert. Tracking Deformable Surfaces with Optical Flow in the Presence of Self-Occlusions in Monocular Image Sequences. In *CVPR Workshops, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, pages 1–6. IEEE Computer Society, June 2008. 7, 8, 45

[HE09a] A. Hilsmann and P. Eisert. Joint Estimation of Deformable Motion and Photometric Parameters in Single View Video. In *ICCV Workshops, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, pages 390–397. IEEE Computer Society, Sept. 2009. 7, 8, 34, 44

[HE09b] A. Hilsmann and P. Eisert. Realistic Cloth Augmentation in Single View Video. In *Proc. Int. Workshop on Vision, Modeling, and Visualization (VMV)*, pages 55–62. DNB, Nov. 2009. 8, 34

[HE09c] A. Hilsmann and P. Eisert. Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications. In *Proc. Int. Conf. on Computer Vision/Computer Graphics Collaboration Techniques and Applications–Mirage*, pages 94–105. Springer, May 2009. 8, 44, 45

[HE12] A. Hilsmann and P. Eisert. Image-Based Animation of Clothes. In *Eurographics Short Papers*, pages 69–72. Eurographics Association, May 2012. 7, 8, 55

[HF01] H. W. Haussecker and D. J. Fleet. Computing Optical Flow with Physical Models of Brightness Variation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):661–673, June 2001. 12

[HFE13] A. Hilsmann, P. Fechteler, and P. Eisert. Pose Space Image Based Rendering. *Comput. Graph. Forum (Proc. Eurographics)*, 32(2):265–274, May 2013. 7, 8, 55

[HG09] M. Hofmann and D. M. Gavrila. Multi-View 3D Human Pose Estimation Combining Single-Frame Recovery, Temporal Integration and Model Adaptation. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2214–2221. IEEE Computer Society, June 2009. 59

[HHS09] P. Huang, A. Hilton, and J. Starck. Human Motion Synthesis from 3D Video. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1478–1485. IEEE Computer Society, June 2009. 15

[HLEL06] J. H. Hays, M. Leordeanu, A. A. Efros, and Y. Liu. Discovering Texture Regularity as a Higher-Order Correspondence Problem. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 522–535. Springer, May 2006. 20

[HMT00] D. C. Hoaglin, F. Mosteller, and J. W. Tukey. *Understanding Robust and Exploratory Data Analysis*. Wiley-Interscience, Jan. 2000. 42

[HO92] P. C. Hansen and D. P. O'Leary. The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems. *SIAM Journ. Sci. Comput.*, 14(6):1487–1503, Nov. 1992. 51

[Hor86] B. K. P. Horn. *Robot Vision*. McGraw-Hill Higher Education, 1st edition, March 1986. 10, 11, 24, 28

[HS81] B. K. P. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185–203, Aug. 1981. 11, 24, 29

[HS88] C. Harris and M. Stephens. A Combined Corner and Edge Detection. In *Proc. 4th Alvey Vision Conf.*, pages 147–151. BMVA, Aug. 1988. 10

[HS07] H. Hirschmüller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE Computer Society, June 2007. 47, 116, 117

[HSE10] A. Hilsmann, D. C. Schneider, and P. Eisert. Realistic Cloth Augmentation in Single View under Occlusion. *Computers & Graphics*, 34(5):567–574, Oct. 2010. 7, 8, 34, 44, 45, 93

[HSE11a] A. Hilsmann, D. C. Schneider, and P. Eisert. Image-Based Retexturing of Deformed Surfaces from a Single Image. In *Eurographics Posters*, pages 43–44. Eurographics Association, April 2011. 7, 8, 93

[HSE11b] A. Hilsmann, D. C. Schneider, and P. Eisert. Template-free Shape-from-Texture with Perspective Cameras. In *Proc. British Machine Vision Conference (BMVC)*, pages 26:1–26:10. BMVA, Sept. 2011. 19, 20, 46, 98, 107, 108

[HSE11c] A. Hilsmann, D. C. Schneider, and P. Eisert. Warp-Based Near-Regular Texture Analysis for Image-Based Texture Overlay. In *Proc. Int. Workshop on Vision, Modeling, and Visualization (VMV)*, pages 73–80. Eurographics Association, Oct. 2011. 7, 8, 93

[HSR11a] S. Hauswiesner, M. Straka, and G. Reitmayr. Free Viewpoint Virtual Try-On With Commodity Depth Cameras. In *Proc. Int. Conf. on Virtual Reality Continuum and Its Applications in Industry (VRCAI)*, pages 23–30. ACM, Dec. 2011. 16, 22

[HSR11b] S. Hauswiesner, M. Straka, and G. Reitmayr. Image-Based Clothes Transfer. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 169–172. IEEE Computer Society, Oct. 2011. 15, 16, 22

[HSR13] S. Hauswiesner, M. Straka, and G. Reitmayr. Virtual Try-On Through Image-based Rendering. *IEEE Trans. on Visualization and Computer Graphics*, 19(9):1552–1565, Sept. 2013. 15, 16, 22

[Hub81] P. J. Huber. *Robust Statistics.* John Wiley & Sons, April 1981. 11, 26, 27

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2nd edition, March 2004. 63

[IA99] M. Irani and P. Anandan. About Direct Methods. In *ICCV Workshops, Workshop on Vision Algorithms: Theory and Practice*, pages 267–277. Springer, Sept. 1999. 10, 24

[IH93] B. Igleicz and D. Hoaglin. *How to Detect and Handle Outliers*, volume 16 of *The ASQC Basic References in Quality Control: Statistical Techniques*. ASQC Quality Press, Jan. 1993. 43, 100, 114

[IMH05] T. Igarashi, T. Moscovich, and J. F. Hughes. As-Rigid-As-Possible Shape Manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3):1134–1141, July 2005. 15

[KJP02] P. G. Kry, D. J. James, and D. K. Pai. EigenSkin: Real Time Large Deformation Character Skinning in Hardware. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 153–159. ACM, July 2002. 17

[KM04] T. Kurihara and N. Miyata. Modeling Deformable Human Hands from Medical Images. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 355–363. Eurographics Association, Aug. 2004. 17

[KU03] J. Kybic and M. Unser. Fast Parametric Elastic Image Registration. *IEEE Trans. Img. Proc.*, 12(11):1427–1442, Nov. 2003. 9

[KV08] T.-Y. Kim and E. Vendrovsky. DrivenShape - a Data-Driven Approach for Shape Deformation. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 49–55. Eurographics Association, July 2008. 18

[LCF00] J. P. Lewis, M. Cordner, and N. Fong. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 165–172. ACM, Aug. 2000. 16, 17, 55, 60, 61, 68, 83, 86

[Lev44] K. Levenberg. A Method for the Solution of Certain Problems in Least Squares. *Quart. Appl. Math.*, 2:164–168, July 1944. 26

[LF06] A. Lobay and D. A. Forsyth. Shape from Texture without Boundaries. *Int. Journ. Comput. Vision*, 67(1):71–91, April 2006. 19, 20, 96, 98

[LH96] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 31–42. ACM, Aug. 1996. 3, 13

[LH05] A. M. Loh and R. Hartley. Shape from Non-Homogeneous, Non-Stationary, Anisotropic, Prespective Texture. In *Proc. British Machine Vision Conf. (BMVC)*, pages 69–78. BMVA, Sept. 2005. 98

[LI07] V. S. Lempitsky and D. V. Ivanov. Seamless Mosaicing of Image-Based Texture Maps. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, June 2007. 13

[LK81] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 674–679. Morgan Kaufmann Publishers Inc., Aug. 1981. 11

[LL03] Y. Liu and W.-C. Lin. Deformable Texture: The Irregular-Regular-Irregular Cycle. Technical report, Robotics Institute, Carnegie Mellon University, Aug. 2003. 94

[LLB+10] C. Lipski, C. Linz, K. Berger, A. Sellent, and M. Magnor. Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time. *Comput. Graph. Forum*, 29(8):2555–2568, Dec. 2010. 14

[LLH04] Y. Liu, W.-C. Lin, and J. Hays. Near-Regular Texture Analysis and Manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):368–376, Aug. 2004. xi, 19, 20, 94, 95, 98

[Low03] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journ. Comput. Vision*, 60(2):91–110, Nov. 2003. 10, 47, 64, 97, 103

[LSH+11] P. Li, H. Sun, C. Huang, J. Shen, and Y. Nie. Efficient Image/Video Retexturing using Parallel Bilateral Grids. In *Proc. Int. Conf. on Virtual Reality Continuum and Its Applications in Industry (VRCAI)*, pages 131–140. ACM, Dec. 2011. 18, 19

[LTL05] Y. Liu, Y. Tsin, and W.-C. Lin. The Promise and Perils of Near-Regular Texture. *Int. Journ. Comput. Vision*, 62(1-2):145–159, April 2005. 19, 95

[LY05] J. Lim and M.-H. Yang. A Direct Method for Modeling Non-Rigid Motion with Thin Plate Spline. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1196–1202. IEEE Computer Society, June 2005. 9, 10, 24

[LYW+10] J. Li, J. Ye, Y. Wang, L. Bai, and G. Lu. Fitting 3D Garment Models onto Individual Human Models. *Computers & Graphics*, 34(6):742 – 755, Dec. 2010. 55

[Mar63] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journ. Appl. Math.*, 11(2):431–441, June 1963. 26

[MB95] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. In *Proc. 22nd Annual Conf. on Computer Graphics and Interactive Techniques (Proc. SIGGRAPH)*, pages 39–46. ACM, Aug. 1995. 13

[MBT+12] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. Data-Driven Estimation of Cloth Simulation Models. *Comput. Graph. Forum (Proc. Eurographics)*, 31(2):519–528, May 2012. 3

[MBW07] Y. Mileva, A. Bruhn, and J. Weickert. Illumination-Robust Variational Optical Flow with Photometric Invariants. In *Proc. 29th DAGM Conf. on Pattern Recognition*, pages 152–162. Springer-Verlag, Sept. 2007. 11

[MCF10] J. Molnár, D. Chetverikov, and S. Fazekas. Illumination-Robust Variational Optical Flow using Cross-Correlation. *Comput. Vis. Image Underst.*, 114(10):1104–1114, Oct. 2010. 12

[Mida] Middlebury Multi-View Stereo Evaluation. http://vision.middlebury.edu/mview. 13

[Midb] Middlebury Stereo Evaluation. http://vision.middlebury.edu/stereo. 13

[MN89] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall, 2nd edition, Aug. 1989. 26

[MNT04] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for Non-Linear Least Squares Problems (2nd ed.)*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004. 25, 26

[MTKV+11] N. Magnenat-Thalmann, B. Kevelham, P. Volino, M. Kasap, and E. Lyard. 3D Web-Based Virtual Try On of Physically Simulated Clothes. *Computer-Aided Design & Applications*, 8(2):163–174, April 2011. 5, 21

[MY09] J.-M. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journ. Img. Sci.*, 2(2):438–469, April 2009. 10

[NISA06] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian Mesh Optimization. In *Proc. Int. Conf. on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE)*, pages 381–389. ACM, Nov. 2006. 32, 33

[NVH+13] T. Neumann, K. Varanasi, N. Hasler, M. Wacker, M. Magnor, and C. Theobalt. Capture and Statistical Modeling of Arm-Muscle Deformations . *Comput. Graph. Forum (Proc. Eurographics)*, 32(2):285–294, May 2013. 16, 17

[NW00] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Aug. 2000. 25, 26

[PBCL09] M. Park, K. Brocklehurst, R. Collins, and Y. Liu. Deformed Lattice Detection in Real-World Images using Mean-Shift Belief Propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(1):1804–1816, Oct. 2009. 20, 95, 96, 98

[PCBC10] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global Solutions of Variational Models with Convex Regularization. *SIAM J. Img. Sci.*, 3(4):1122–1145, Dec 2010. 11, 12

[PDG05] D. Porquet, J.-M. Dischler, and D. Ghazanfarpour. Real-Time High-Quality View-Dependent Texture Mapping using Per-Pixel Visibility. In *Proc. Int. Conf. on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)*, pages 213–220. ACM, Nov. 2005. 13

[PGB03] P. Pérez, M. Gangnet, and A. Blake. Poisson Image Editing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 22(3):313–318, July 2003. 114

[PLF05a] J. Pilet, V. Lepetit, and P. Fua. Augmenting Deformable Objects in Real-Time. In *Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 134–137. IEEE Computer Society, Oct. 2005. 9, 18

[PLF05b] J. Pilet, V. Lepetit, and P. Fua. Real-Time Non-Rigid Surface Detection. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 822–828. IEEE Computer Society, June 2005. 18, 19

[PLF08] J. Pilet, V. Lepetit, and P. Fua. Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. *Int. Journ. Comput. Vision*, 76(2):109–122, Jan. 2008. 10, 18, 19, 30

[PSG+08] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A Convex Formulation of Continuous Multi-Label Problems. In *Proc. Europ. Conf. on Computer Vision (ECCV*, pages 792–805. Springer, Oct. 2008. 11

[PZB+09] T. Popa, Q. Zhou, D. Bradley, V. Kraevoy, H. Fu, A. Sheffer, and W. Heidrich. Wrinkling Captured Garments Using Space-Time Data-Driven Deformation. *Comput. Graph. Forum (Proc. Eurographics)*, 28(2):427–435, April 2009. 3

[RHE11] D. Lopez Recas, A. Hilsmann, and P. Eisert. Near-Regular Texture Synthesis by Random Sampling and Gap Filling. In *Proc. Int. Workshop on Vision, Modeling, and Visualization Workshop (VMV)*, pages 89–96. Eurographics Association, Oct. 2011. 103, 105, 106

[RLN06] T. Rhee, J. P. Lewis, and U. Neumann. Real-Time Weighted Pose-Space Deformation on the GPU. *Comput. Graph. Forum (Proc. Eurographics)*, 25(3):439–448, Sept. 2006. 72

[SCD+06] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528. IEEE Computer Society, June 2006. 13

[SD96] S. Seitz and C. Dyer. View Morphing. In *Proc. 23rd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 21–30. ACM, Aug. 1996. 14

[SD97] S. M. Seitz and C. R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1067–1073. IEEE Computer Society, June 1997. 13

[SDP13] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable Medical Image Registration: A Survey. *IEEE Trans. on Medical Imaging*, 32(7):1153–1190, July 2013. 9

[SFBH09] S. Sedai, F. Flitti, M. Bennamoun, and D. Huynh. 3D Human Pose Estimation from Static Images Using Local Features and Discriminative Learning. In *Proc. Int. Conf. on Image Analysis and Recognition (ICIAR)*, pages 327–336. Springer, July 2009. 59

[SFC+11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304. IEEE Computer Society, June 2011. 5, 21

[She68] D. Shepard. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In *Proc. of the 23rd ACM National Conf.*, pages 517–524. ACM, Aug. 1968. 69

[She96] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer, May 1996. 31, 62

[SHE11a] D. C. Schneider, A. Hilsmann, and P Eisert. A Global Optimization Approach to High-Detail Reconstruction of the Head. In *Proc. Int. Workshop on Vision, Modeling, and Visualization (VMV)*, pages 9–15. Eurographics Association, Oct. 2011. 12, 34

[SHE11b] D. C. Schneider, A. Hilsmann, and P. Eisert. Deformable Image Alignment as a Source of Stereo Correspondences on Portraits. In *CVPR Workshops, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, pages 45–52. IEEE Computer Society, June 2011. 9

[SHE11c] D. C. Schneider, A. Hilsmann, and P. Eisert. Warp-Based Motion Compensation for Endoscopic Kymography. In *Eurographics Short Papers*, pages 41–44. Eurographics Association, April 2011. 9

[SHF07] M. Salzmann, R. Hartley, and P. Fua. Convex Optimization for Deformable Surface 3-D Tracking. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 1–8. IEEE Computer Society, Oct. 2007. 118

[SK00] H. Y. Shum and S. B. Kang. A Review of Image-Based Rendering Techniques. In *Proc. Visual Communications and Image Processing (VCIP)*, pages 1–12. SPIE, Nov. 2000. 13

[SLW+11] T. Stich, C. Linz, C. Wallraven, D. Cunningham, and M. Magnor. Perception-Motivated Interpolation of Image Sequences. *ACM Trans. Appl. Percept.*, 8(2):1–25, Feb. 2011. 14

[SM06] V. Scholz and M. Magnor. Texture Replacement of Garments in Monocular Video Sequences. In *Proc. Eurographics Symp. on Rendering*, pages 305–312. Eurographics Association, June 2006. 10, 18

[SM07] G. Silveira and E. Malis. Real-Time Visual Tracking under Arbitrary Illumination Changes. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6. IEEE Computer Society, June 2007. 12

[SMH05] J. Starck, G. Miller, and A. Hilton. Video-Based Character Animation. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 49–58. ACM, July 2005. 15

[SMNLF08] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-Form Solution to Non-Rigid 3D Surface Registration. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 581–594. Springer, Oct. 2008. 118

[SRB10] D. Sun, S. Roth, and M. J. Black. Secrets of Optical Flow Estimation and Their Principles. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439. IEEE Computer Society, June 2010. 29

[SRC01] P.-P. J. Sloan, C. F. Rose, and M. F. Cohen. Shape by Example. In *Proc. Symp. on Interactive 3D Graphics (I3D)*, pages 135–143. ACM, March 2001. 16, 17, 55, 68

[SSK+05] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment Motion Capture Using Color-Coded Patterns. *Comput. Graph. Forum (Proc. Eurographics)*, 24(3):439–448, March 2005. 18

[SSS08] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *Int. Journ. Comput. Vision*, 80(2):189–210, Nov 2008. 46, 62, 63

[STL08] L. Shen, P. Tan, and S. Lin. Intrinsic Image Decomposition with Non-Local Texture Cues. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7. IEEE Computer Society, June 2008. 12

[Tau95] G. Taubin. A Signal Processing Approach to Fair Surface Design. In *Proc. 22nd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 351–358. ACM, Aug. 1995. 32, 33

[TC07] T. W. H. Tang and A. C. S. Chung. Non-Rigid Image Registration using Graph-Cuts. In *Pro. Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 916–924. Springer-Verlag, Nov. 2007. 12

[TGSY95] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Kluwer Academic Publishers, June 1995. 26, 28

[Thi96] J. Thirion. New Feature Points Based on Geometric Invariants for 3D Image Registration. *Int. Journ. Comput. Vision*, 18(2):121–137, May 1996. 10

[TLCH02] C.-H. Teng, S.-H. Lai, Y.-S. Chen, and W.-H. Hsu. Robust Computation of Optical Flow under Non-Uniform Illumination Variations. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 327–330. IEEE Computer Society, Aug. 2002. 12

[TLR01] Y. Tsin, Y. Liu, and V. Ramesh. Texture Replacement in Real Images. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 539 – 544. IEEE Computer Society, Dec. 2001. 20

[TS09] H. Tanaka and H. Saito. Texture Overlay onto Flexible Objects with PCA of Silhouettes and k-Means Methods for Search into Database. In *Proc. IAPR Conf. on Machine Vision Applications (MVA)*, pages 5–8. Springer, May 2009. 16, 22

[VBK02] S. Vedula, S. Baker, and T. Kanade. Spatio-Temporal View Interpolation. In *Proc. Eurographics Workshop on Rendering*, pages 65–76. ACM, June 2002. 14

[vdWG04] J. van de Weijer and T. Gevers. Robust Optical Flow from Photometric Invariants. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 1835–1838. IEEE Computer Society, Oct. 2004. 11

[VGB06] L. Vanaken, M. Gerrits, and P. Bekaert. Animated Video Sprites. In *Eurographics Short Papers*, pages 69–71. Eurographics Association, Sept. 2006. 15

[VMTF09] P. Volino, N. Magnenat-Thalmann, and F. Faure. A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation. *ACM Trans. Graph.*, 28(4):105:1–105:16, Aug. 2009. 3

[VSFU12] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A Constrained Latent Variable Model. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2248–2255. IEEE Computer Society, June 2012. 118

[WCF07] R. White, K. Crane, and D. A. Forsyth. Capturing and Animating Occluded Cloth. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):34:1–34:8, Aug. 2007. 10, 18

[Wed74] R. W. M. Wedderburn. Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. *Biometrika*, 61(3):439–447, Dec. 1974. 26

[WF06] R. White and D. A. Forsyth. Retexturing Single Views Using Texture and Shading. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 70–81. Springer, May 2006. 18

[WHRO10] H. Wang, F. Hecht, R. Ramamoorthi, and J. O'Brien. Example-Based Wrinkle Synthesis for Clothing Animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29(4):107:1–107:8, July 2010. 4, 16, 17, 18, 55, 56, 59, 72

[WKK+05] M. Wacker, M. Keckeisen, S. Kimmerle, W. Straßer, V. Luckas, C. Groß, A. Fuhrmann, R. Sarlette, M. Sattler, and R. Klein. Simulation and Visualisation of Virtual Textiles for Virtual Try-On. *Journ. of Textile and Apparel: Virtual Clothing Technology and Applications*, 9(1):37–47, 2005. 21

[WMKG07] M. Wardetzky, S. Mathurand, F. Kälberer, and E. Grinspun. Discrete Laplace Operators: No Free Lunch. In *Proc. Eurographics Symp. on Geometry Processing*, pages 33–37. Eurographics Association, July 2007. 32, 33

[WPZ+09] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An Improved Algorithm for TV-L1 Optical Flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis, Lecture Notes in Computer Science*, pages 23–45. Springer, July 2009. 11, 12

[WSLG07] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman. Context-Aware Skeletal Shape Deformation. *Comput. Graph. Forum (Proc. Eurographics)*, 26(3):265–273, Sept. 2007. 16, 17, 59

[WWG07] M. Waschbüsch, S. Würmlin, and M. H. Gross. 3D Video Billboard Clouds. *Comput. Graph. Forum (Proc. Eurographics)*, 26(3):561–569, June 2007. 14

[XJM10] L. Xu, J. Jia, and Y. Matsushita. Motion Detail Preserving Optical Flow Estimation. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1293–1300. IEEE Computer Society, June 2010. 10, 11

[XLS+11] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt. Video-Based Characters - Creating New Human Performances from a Multi-View Video Database. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 30(4):32:1–32:10, July 2011. 15, 59

[XRS02] J. Xiao, C. Rao, and M. Shah. View Interpolation for Dynamic Scenes. In *Eurographics Short Papers*. Eurographics Association, Sept. 2002. 15

[YS10] X. Yan and J. Shen. Mesh-Guided Texture Replacement using Intrinsic Images. In *Proc. Int. Conf. on Progress in Informatics and Computing (PIC)*, pages 701 –705. IEEE Computer Society, Dec. 2010. 18, 19

[YXLX11] X. Yang, Z. Xue, X. Liu, and D. Xiong. Topology Preservation Evaluation of Compact-Support Radial Basis Functions for Image Registration. *Pattern Recogn. Lett.*, 32(8):1162–1177, June 2011. 9

[ZF03] B. Zitova and J. Flusser. Image Registration Methods: A Survey. *Image and Vision Computing*, 21(11):977–1000, Oct. 2003. 9

[ZGK+10] D. Zikic, B. Glocker, O. Kutter, M. Groher, N. Komodakis, A. Kamen, N. Paragios, and N. Navab. Linear Intensity-Based Image Registration by Markov Random Fields and Discrete Optimization. *Medical Image Analysis*, 14(4):550–562, March 2010. 11, 12

[ZKU+04] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-Quality Video View Interpolation using a Layered Representation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):600–608, Aug. 2004. 14

[ZLMH09] J. Zhu, M. R. Lyu, R. Michael, and T. S. Huang. A Fast 2D Shape Recovery Approach by Fusing Features and Appearance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(7):1210–1224, July 2009. 9, 11, 12

[ZSZ+12] Z. Zhou, B. Shu, S. Zhuo, X. Deng, P. Tan, and S. Lin. Image-Based Clothes Animation for Virtual Fitting. In *ACM SIGGRAPH Asia 2012 Technical Briefs*, pages 33:1–33:4. ACM, Nov. 2012. 16, 22

[ZTCS99] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from Shading: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, Aug. 1999. 19